HAW HAMBURG

BACHELOR THESIS
Tom Florentinus Bachmann

Detektion von Korkporen an Äpfeln mittels Bildverarbeitung und ggf. Maschinellem Lernen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Tom Florentinus Bachmann

Detektion von Korkporen an Äpfeln mittels Bildverarbeitung und ggf. Maschinellem Lernen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Peer Stelldinger
Zweitgutachterin: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 16.09.2024

Tom Florentinus Bachmann

Thema der Arbeit

Detektion von Korkporen an Äpfeln mittels Bildverarbeitung und ggf. Maschinellem Lernen

Stichworte

Korkporen, Bildverarbeitung, Maschinelles Lernen

Kurzzusammenfassung

Die Arbeit befasst sich mit dem Nachweis von Korkporen auf Äpfeln. Dazu werden verschiedene Ansätze verwendet und miteinander verglichen. Es zeigt sich, dass verschiedene Arten von CNNs den klassischen Bildverarbeitungsmethoden wie Blob-Detektor oder Erkennung lokaler Extrema überlegen sind. Zusätzlich wurden fünf Modelle, Faster R-CNN, RetinaNet, SSD, U-Net und YOLO, verglichen. Es zeigt sich, dass der Datensatz, mit dem die Modelle trainiert wurden, zwar groß genug ist, um Kokrporen auf Äpfeln zu erkennen, aber nicht vielfältig genug. So haben viele Modelle Schwierigkeiten mit unbekannten Situationen, wie z.B. fremden Sorten oder Äpfeln im Hintergrund. Die vielversprechendsten Ergebnisse liefern U-Net und YOLO, da sie im Vergleich zu den anderen Modellen robuster sind. Für SSD hingegen war der Datensatz zu klein, so dass das Modell keine Korkporen erkennen konnte. Basierend auf diesen Ergebnissen ist es möglich, eine hohe Erkennungsrate von Korkporen auf Äpfeln zu erreichen, aber es müssen noch einige Optimierungen vorgenommen werden.

Tom Florentinus Bachmann

Title of Thesis

Detection of corkpores on apples through Image Processing and Machine Learning

Keywords

Corkpores, Image Processing, Machine Learning

Abstract

The work deals with the detection of cork pores on apples. Different approaches are used and compared with each other. It is shown that different types of CNNs are superior to classical image processing methods such as blob detector or detection of local extrema. In addition, five models, Faster R-CNN, RetinaNet, SSD, U-Net and YOLO, were compared. It turns out that the data set used to train the models is large enough to recognise cocrpores on apples, but not diverse enough. For example, many models have difficulties with unknown situations, such as foreign varieties or apples in the background. The most promising results are provided by U-Net and YOLO, as they are more robust compared to the other models. For SSD, on the other hand, the data set was too small, so the model could not recognise cork pores. Based on these results, it is possible to achieve a high detection rate of cork pores on apples, but some optimisations still need to be made.

Inhaltsverzeichnis

Abbildungsverzeichnis **vii**

Tabellenverzeichnis **ix**

Abkürzungen **x**

1 Einleitung **1**
1.1 Einordnung der Arbeit . . . 2
1.2 Aufgabe der Arbeit . . . 2

2 Grundlagen **4**
2.1 Korkporen auf Äpfel . . . 4
2.2 Bekannte Forschungen . . . 5

3 Traditionelle Bildverarbeitung **7**
3.1 Thresholding . . . 7
3.1.1 Grundprinzipien des Thresholding . . . 8
3.1.2 Ergebnisse . . . 8
3.2 Lokales Maximum/Minimum . . . 10
3.2.1 Grundprinzipien der Detektion . . . 10
3.2.2 Ergebnisse . . . 11
3.3 Blob-Detektor . . . 12
3.3.1 Grundprinzipien der Blob-Detektion . . . 12
3.3.2 Ansätze zur Detektion von Blobs . . . 13
3.3.3 Ergebnisse . . . 13
3.4 Fazit . . . 14

4 Neuronale Netze **15**
4.1 Modellwahl . . . 15
4.1.1 One-Staged Modelle (YOLO, SSD, RetinaNet) . . . 16

4.1.2 Two-Staged Modelle (R-CNN) . . . 19
4.1.3 Segmentierungsmodelle (U-Net) . . . 20
4.1.4 Fazit . . . 21
4.2 Versuchsaufbau . . . 22
4.3 Datensatz . . . 23
4.4 Labelisierung . . . 23
4.5 Vorbereitung und Arten von Testbildern . . . 25
4.5.1 Rauschen . . . 27
4.5.2 Runterskalierung (Downsampling) . . . 30

5 Ergebnisanalyse 33
5.1 Statistik . . . 34
5.1.1 Mean Average Precision (mAP) . . . 34
5.1.2 Loss . . . 36
5.2 Apfelsorten . . . 38
5.2.1 Trainierte Apfelsorten . . . 38
5.2.2 Unbekannte Apfelsorten . . . 43
5.3 Bildqualität . . . 48
5.3.1 Rauschen . . . 49
5.3.2 Downsampling . . . 53
5.4 Beleuchtung . . . 54
5.5 Hintergrund . . . 56
5.6 Defekte . . . 58
5.7 Oben/Unten . . . 58
5.7.1 Oben . . . 59
5.7.2 Unten . . . 60

6 Fazit 62
6.1 Zusammenfassung . . . 62
6.2 Einordnung (Nachfolgearbeiten) . . . 64

Literaturverzeichnis 65

A Anhang 69
A.1 Verwendete Hilfsmittel . . . 69

Selbstständigkeitserklärung 70

Abbildungsverzeichnis

3.1 Detektion mittels Einfachem Thresholding 9
3.2 Detektion mittels Adaptivem Thresholding 10
3.3 Detektion mittels lokaler Minima 12
3.4 Detektion mittels Blob-Detektor 14

4.1 Auswahl an Bildern aus dem Datensatz 24
4.2 Interface des Tools zum Erstellen der Labels 25
4.3 Vergleich der Bilder mit Gausschem Rauschen 28
4.4 Vergleich der Bilder mit Salz-und-Pfeffer-Rauschen 29
4.5 Vergleich der Bilder nach Anwendung von Downsampling 31
4.6 Auswahl an Bildern aus dem Testdatensatz 32

5.1 Mean Average Precision (mAP) 35
5.2 Loss 37
5.3 Golden Delicious 39
5.4 'Bekannter' Wellant 41
5.5 'Fremder' Wellant 42
5.6 Crimson Snow 44
5.7 Granny Smith 45
5.8 Elstar 47
5.9 Red Jonaprince 48
5.10 U-Net mit verschieden starken Standardabweichungen 50
5.11 You Only Look Once (YOLO) mit verschieden starken Standardabweichungen 51
5.12 Salz-und-Pfeffer Rauschen mit einer Rauschdichte von 0.1 52
5.13 Herunterskalierung mit einem Faktor von $\frac{1}{8}$ 54
5.14 Stark beleuchteter Wellant 55
5.15 Wellant aus gleicher Position mit besser erkennbaren Korkporen 56
5.16 Apfel mit Hintergrund 57

5.17 Defekte 59
5.18 Oben 60
5.19 Unten 61

Tabellenverzeichnis

A.1 Verwendete Hilfsmittel und Werkzeuge 69

Abkürzungen

BMEL Bundesministerium für Ernährung und Landwirtschaft.

CNN Convolutional Neural Network.

DNN Deep Neural Network.

Faster R-CNN Faster Region-based Convolutional Neural Network.

FPN Feature Pyramid Network.

IoU Intersection of Union.

mAP Mean Average Precision.

R-CNN Region-based Convolutional Neural Network.

RPN Region Proposal Networks.

SAMSON Smart Agricultural Monitoring and Optimization System.

SSD Single Shot Multibox Detector.

YOLO You Only Look Once.

1 Einleitung

Die Apfelproduktion spielt im Alten Land eine zentrale Rolle und genießt laut Kleefeld, Burggraaff und Lange [2007] national und international einen hervorragenden Ruf. Jährlich werden mehrere hunderttausend Tonnen Äpfel geerntet[1], womit das Alte Land einen Großteil der in Deutschland geernteten Äpfel liefert[2].

Neben der traditionellen Anbauweise spielt in der Apfelproduktion auch die Forschung und Entwicklung eine immer größere Rolle. Moderne Technologien und innovative Ansätze werden für die Qualitätskontrolle und Ertragssteigerung immer relevanter. Gerade im Hinblick auf zukünftige Herausforderungen, wie den Klimawandel, werden nachhaltige und effiziente Methoden im Apfelanbau entscheidend sein.

Diese Bachelorarbeit knüpft an diese Herausforderung an und beschäftigt sich mit der Optimierung des Apfelanbaus im Alten Land. Sie beschäftigt sich insbesondere mit der Erkennung von Korkporen auf Äpfeln. Die gewonnenen Erkenntnisse tragen dazu bei, die Qualität des Erntegutes zu verbessern, indem sie die Grundlage dafür bilden, jeden Apfel während seiner Entwicklung zu verfolgen. So wird durch die Verbindung von vorausgegangener Forschung und neuen Technologien sichergestellt, dass das Alte Land eine bedeutende Produktionsstätte in Europa bleibt.

[1] Laut Norddeutschem Rundfunk wurden im Jahr 2023 ca. 300.000 Tonnen Äpfel geerntet.
https://www.ndr.de/fernsehen/sendungen/hallo_niedersachsen/Altes-Land-Ernte-von-300000-Tonnen-Aepfel-hat-begonnen,hallonds82322.html#:~:text=Jeder%20dritte%20Apfel%20in%20der%20Bundesrepublik%20kommt%20aus%20dem%20Alten%20Land (aufgerufen am 26.06.2024)

[2] Laut Statistischem Bundesamt wurden in Deutschland im Jahr 2023 941.200 Tonnen Äpfel geerntet. Daher macht die Apfelernte im Alten Land ungefähr ein Drittel aller in Deutschland geernteten Äpfel aus.
https://www.destatis.de/DE/Presse/Pressemitteilungen/2024/01/PD24_008_412.html#:~:text=Wie%20das%20Statistische%20Bundesamt%20(Destatis,1%2C1%20Millionen%20Tonnen) (aufgerufen am 28.06.2024)

1.1 Einordnung der Arbeit

Die Arbeiten sind Teil des vom Bundesministerium für Ernährung und Landwirtschaft (BMEL) geförderten Forschungsprojektes Smart Agricultural Monitoring and Optimization System (SAMSON)[3], das mit modernen Technologien und Konzepten eine nachhaltigere Landwirtschaft fördern will. SAMSON erforscht die Entwicklung von Automatisierungssystemen mit künstlicher Intelligenz. Diese unterstützen Obstbauern bei der Überwachung und Datenerfassung. So kann beispielsweise die Qualität und Quantität von Äpfeln über mehrere Jahre hinweg verglichen und die Durchführung von Maßnahmen vereinfacht werden.

Im Rahmen dieses Projektes geht es um die zuverlässige Erkennung von Korkporen auf Äpfeln. Die genaue Erkennung ist ein wichtiger Grundstein für die Verbesserung der Qualitätskontrolle und damit für eine bessere Ernte. Verschiedene Methoden der traditionellen Bildverarbeitung und des maschinellen Lernens werden verglichen, um die effektivste und zuverlässigste Technik zur Erkennung von Korkporen zu ermitteln.

Für den Apfelanbau ergeben sich mehrere Vorteile. So bietet eine genaue Erkennung der Korkporen die Möglichkeit, Äpfel zu identifizieren und damit über ihre gesamte Entwicklung zurückzuverfolgen. Dies ermöglicht eine lückenlose Überwachung des Wachstumsprozesses, was sowohl für den Obstbauern als auch für den Verbraucher von Vorteil ist. Obstbauern können einen aussortierten Apfel bis zu seinem Ursprung zurückverfolgen, was die Behebung von Qualitätsproblemen erleichtert. Ebenso sorgt die Rückverfolgbarkeit der Lieferkette für Transparenz, was in der heutigen Zeit eine immer wichtigere Rolle spielt.

Vorangegangen an diese Arbeit wurde bereits eine Bachelorarbeit zur Apfelidentifikation anhand simulierter Korkporen von Hauke Berner verfasst. Diese hat einen wichtigen Beitrag zur Forschung an diesem Projekt beigetragen und bietet für diese Arbeit einige Anhaltspunkte.

1.2 Aufgabe der Arbeit

In dieser Arbeit geht es darum, verschiedene Ansätze zur Detektion von Korkporen auf Äpfeln zu testen und deren Performance zu vergleichen. Dabei sollen sowohl traditionelle

[3] https://samson-projekt.de/projektbeschreibung/ (aufgerufen am 28.06.2024)

Bildverarbeitungsmethoden als auch moderne Variationen von Convolutional Neural Network (CNN)s eingesetzt werden. Primär geht es bei dem Vergleich um eine zuverlässige Methode zur Identifikation der Korkporen.

Ein wichtiger Aspekt der Arbeit ist die Benutzung von CNNs. Diese haben sich in der Bildverarbeitung und -analyse etabliert und gelten als der Standard für Objekterkennung[4],[5]. Durch ihre tiefe Netzstruktur bieten sie die Möglichkeit, auch komplexe Muster und Strukturen in Bildern zu erkennen. In dieser Arbeit werden verschiedene Modelle getestet, um die bestmögliche Korkporenerkennung zu erzielen.

Dabei fokussiert sie sich lediglich auf die Detektion von Korkporen. Weiterführende Schritte wären zum Beispiel die Erfassung von der Position der Korkporen und deren Anordnung untereinander oder sogar die Entwicklung einer dreidimensionalen Nachbildung des Apfels anhand seiner Korkporen. Diese Möglichkeiten können auf dieser Arbeit aufbauen, sind aber kein Teil davon. Auch die Identifikation eines Apfels anhand seiner Korkporen soll hier nicht behandelt werden.

So soll durch den Vergleich eine fundierte Grundlage für die Identifikation und Verfolgung von Äpfeln geschaffen werden. Die Ergebnisse dieser Arbeit dienen als ein erster Ansatz in diese Richtung.

Insgesamt liefert die Arbeit einen neuen Beitrag zur Optimierung des Apfelanbaus im Alten Land, indem moderne Technik, wie trainierte Maschinen, in die Landwirtschaft integriert werden und so dabei hilft, die Qualität der Ernte zu sichern und den Apfelanbau gegen zukünftigen Herausforderungen zu wappnen.

[4] https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns (aufgerufen am 30.06.2024)
[5] https://www.opendatascience.com/using-the-cnn-architecture-in-image-processing/ (aufgerufen am 30.06.2024)

2 Grundlagen

Die Grundlagen sollen einen Überblick über die für diese Arbeit notwendigen Teilgebiete geben. Aus diesem Grund wird zunächst ausführlich auf die Grundlagen der Korkporen eingegangen. Diese bilden den Kern der Arbeit und sind daher essentiell. Desweiteren werden bekannte Forschungsarbeiten auf diesem Gebiet angeführt, um den aktuellen Stand der Forschung zu verdeutlichen.

2.1 Korkporen auf Äpfel

Die oberste Schicht der Apfelschale ist von kleinen Öffnungen, wie Rissen, Spalten oder Punkten durchzogen. Deshalb ist kein Apfel wie der andere. Jeder bildet seine eigenen Merkmale aus, die die darunter liegenden Schichten beeinflussen. Die Punkte, die je nach Apfelsorte unterschiedlich gefärbt sein können, nennt man Korkporen oder Lentizellen. Korkporen finden sich nicht nur auf Äpfeln, sondern auch auf der Oberfläche anderer Pflanzen, wie z.B. auf Baumrinde. Im Folgenden wird entsprechend dem Thema der Arbeit ausschließlich auf Korkporen auf Äpfeln eingegangen.

In den ersten Monaten des Wachstumsprozesses bilden sich auf der Schale des Apfels kleine Haaransatzstellen, die sich nach etwa zwei Monaten zu Korkporen weiterentwickeln. Laut Krapf [1961] bilden sich Korkporen im Allgemeinen vor allem dort, wo der Apfel anfällig für äußere Einflüsse ist. So entwickeln sie sich auch aus defekten Hautstellen oder Spaltöffnungen der Epidermis, die direkt unter der Schale liegt. Die Verfärbung der Korkporen ist darauf zurückzuführen, dass die darunter liegenden Zellen nach ihrer Bildung absterben, damit von der Versorgung abgeschnitten werden und so den Wasserverlust regulieren (Khanal, Si und Knoche [2020]).

Die Hauptfunktion der Korkporen auf Äpfeln besteht darin, den Gasaustausch zu ermöglichen. Dies ist entscheidend für die Atmung der Frucht, bei der Sauerstoff gegen Kohlendioxid ausgetauscht wird, sowie für die Transpiration, bei der Wasser durch die

Korkporen verdunstet. Dieser Prozess hilft der Frucht, überschüssige Wärme abzuführen und die Aufnahme von Nährstoffen zu regulieren.

Khanal, Si und Knoche [2020] haben gezeigt, dass die Korkporen zur Transpiration beitragen. Unter Transpiration versteht man die Verdunstung von Wasser. Dies geschieht durch die Öffnungen einer Pflanze. Bei Äpfeln sind dies unter anderem die Korkporen, die je nach Größe mehr oder weniger Wasserdampf durchlassen.

Die Korkporen sind also ein wichtiger Bestandteil des Apfels, da sie qualitäts- und lagerrelevante Funktionen übernehmen. Darüber hinaus ist ihre Anordnung zufällig über den Apfel verteilt, so dass ein Apfel durch die Verteilung seiner Korkporen mit hoher Wahrscheinlichkeit eindeutig identifiziert werden kann. Dies ist besonders wichtig, da mit dem Wissen aus dieser Arbeit ein System zur Identifizierung eines Apfels entwickelt werden kann. Auf diese Weise kann eine fortschrittliche Apfelproduktion gewährleistet werden.

2.2 Bekannte Forschungen

Diese Arbeit beschäftigt sich mit dem Nachweis von Korkporen auf Äpfeln. Dabei sollen verschiedene Ansätze und Methoden miteinander verglichen werden. Während sich bisherige Arbeiten eher mit der Erkennung von Oberflächenfehlern an Früchten beschäftigt haben, soll hier speziell auf Korkporen an Äpfeln eingegangen werden.

Beispielsweise haben Mohana und Prabhakar [2015] die automatische Erkennung von Oberflächenfehlern auf Zitrusfrüchten mit Hilfe von Computer-Vision-Techniken untersucht. Sie zeigten, wie die Bildverarbeitung eingesetzt werden kann, um spezifische Merkmale auf Früchten zu identifizieren. Shermila, Victor, Manoj und Devi [2023] untersuchten auch die Erkennung von Defekten auf Zitrusfrüchten durch die Anwendung von Convolutional Neural Network (CNN)s und zeigten die Vorteile auf, die durch diese Modelle erzielt werden können. Khattak, Asghar, Batool, Zubair, Ullah, Alrakhami und Gumaei [2021] verwendeten Deep Neural Network (DNN)s zur Erkennung von Krankheiten bei Zitrusfrüchten.

Im Bereich der Apfelverarbeitung wurden ebenfalls Bildverarbeitungstechniken untersucht. Allerdings ging es auch hier um die Erkennung von Defekten auf der Oberfläche von Äpfeln. Rehkugler und Throopmann [1989] gehörten zu den ersten, die die Möglichkeiten der automatischen Erkennung von Apfelkrankheiten mittels Bildverarbeitung

untersuchten. Darüber hinaus analysierten Samajpati und Degadwala [2015] herkömmliche Bildverarbeitungsalgorithmen zur Erkennung von Oberflächenfehlern. Eine weitere Arbeit von Azgomi und Haredasht [2022] beschäftigt sich mit maschinellem Lernen. Darüber hinaus haben Guo-zhen, Ming-xing, Ze-xu und Ji-chao [2023] maschinelles Lernen mit der Anwendung von Hyperspektralbildern kombiniert, um Fehler auf Äpfeln zu erkennen.

Derzeit gibt es keine wissenschaftlichen Arbeiten über die Erkennung von Korkporen mit Hilfe von Bildverarbeitung. Diese Arbeit baut jedoch auf den Ergebnissen früherer Arbeiten auf, da sich in den letzten Jahren der Forschung herausgestellt hat, dass sowohl die traditionelle Bildverarbeitung als auch modernere Techniken, wie maschinelles Lernen, der richtige Ansatz für die Erkennung von Oberflächenfehlern sind.

3 Traditionelle Bildverarbeitung

Um das gesamte Spektrum der Bildverarbeitung abzudecken, wurden neben den neuronalen Netzen auch traditionelle Techniken der Bildverarbeitung auf Funktionalität überprüft. Dabei hat sich gezeigt, dass diese gegenüber den trainierten Netzen auch Vorteile haben können. So weisen die klassischen Methoden im Hinblick auf die Rotation eine größere Robustheit gegenüber den CNNs auf, wie Özaydın, Georgiou und Lew [2019] ersichtlich machen. Auch in der Studie von Hegde, Prasad, Hebbar und Singh [2019] wird dargelegt, dass klassische Methoden weniger Ressourcen beanspruchen und keine gelabelten Daten benötigen. Dies unterstreicht die Relevanz, diese Techniken auch in diesem Anwendungsfeld zu evaluieren und hinsichtlich ihrer Performance zu vergleichen. Die in dieser Arbeit verwendeten Techniken, wie der Einsatz von Thresholding, die Erkennung von lokalen Extrema oder Blob-Detektion, werden daher im Folgenden näher beschrieben. Das Verständnis der klassischen Techniken macht deutlich, warum der Fokus auf neuronalen Netzen liegt und welche Vorteile sie bieten. So benötigen herkömmliche Verfahren in der Regel fest definierte Regeln oder Algorithmen, während neuronale Netze durch ihr erlerntes Verhalten auch bei komplexeren Aufgaben deutlich flexibler bleiben.

3.1 Thresholding

Thresholding ist eine grundlegende Methode der Bildverarbeitung, die verwendet wird, um Bilder in Bereiche oder Objekte zu segmentieren. Durch das Setzen eines Schwellenwerts (Threshold) können Pixel, deren Farbwert über oder unter diesem Wert liegt, voneinander getrennt werden. Diese Technik ist besonders nützlich für die Identifizierung und Isolierung von Objekten in einem Bild.

3.1.1 Grundprinzipien des Thresholding

Das Prinzip des Thresholding basiert auf der Festlegung eines oder mehrerer Schwellenwerte, die mit dem Farbwert eines Pixels vergleichen. Je nach Ergebnis des Vergleichs wird der Pixel entweder zu einem Objekt oder zum Hintergrund klassifiziert.

Einfaches Thresholding

Beim einfachen Thresholding wird ein einzelner Schwellenwert T festgelegt. Jeder Pixelwert $I(x, y)$ wird dann mit diesem Schwellenwert verglichen:

$$I'(x,y) = \begin{cases} 255 & \text{wenn } I(x,y) > T \\ 0 & \text{wenn } I(x,y) \leq T \end{cases} \tag{3.1}$$

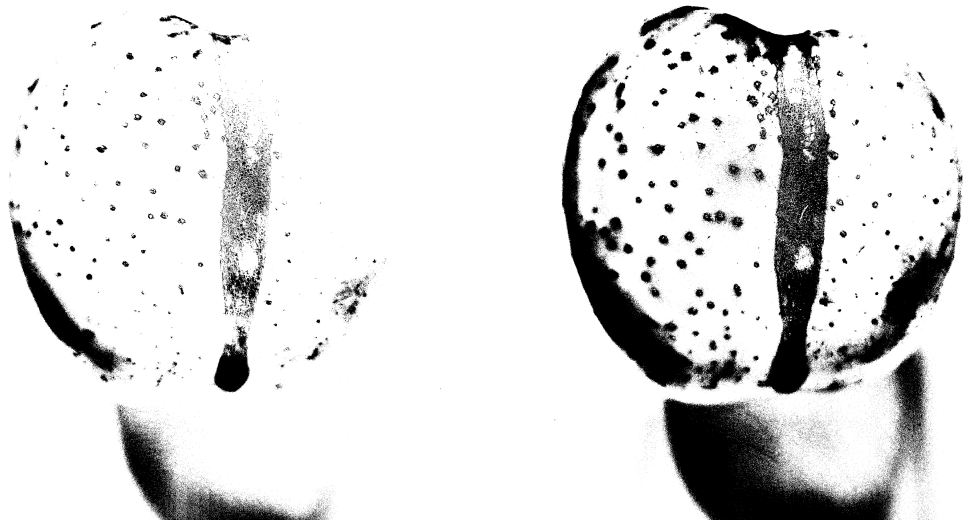
Adaptives Thresholding

Ein einfaches globales Thresholding kann ineffektiv sein, wenn die Beleuchtung im Bild ungleichmäßig ist. In solchen Fällen wird adaptives Thresholding verwendet, das lokale Schwellenwerte anstelle eines globalen Schwellenwerts einsetzt. Der Schwellenwert $T(x, y)$ wird für jedes Pixel basierend auf den Intensitätswerten in seiner Nachbarschaft berechnet. Eine verbreitete Methode ist das Berechnen des Mittelwerts der Nachbarschaftspixel:

$$I'(x,y) = \begin{cases} 255 & \text{wenn } I(x,y) > T(x,y) \\ 0 & \text{wenn } I(x,y) \leq T(x,y) \end{cases} \tag{3.2}$$

3.1.2 Ergebnisse

Das einfache Thresholding liefert, wie in Abbildung 3.1 ersichtlich, keine guten Ergebnisse, da schon kleinste Helligkeitsdifferenzen zu schwachen Ergebnissen führen. Da der Thresholdwert auf das ganze Bild angewendet wird, werden bei verschiedener Helligkeit manche Korkporen übersehen. Wenn der Wert höher gesetzt wird, wird zu wenig herausgefiltert, sodass sehr viel mehr als nur die Korkporen identifiziert werden. Wenn er zu



(a) Korkporen (oben rechts) werden bei niedrigem Threshold übersehen

(b) Zu wenig wird bei hohem Threshold herausgefiltert

Abbildung 3.1: Detektion mittels Einfachem Thresholding

niedrig ist, gehen Korkporen verloren. So ist es nicht möglich, den richtigen Thresholdwert zu identifizieren.

Das adaptive Thresholding greift an dieser Stelle, da es nicht einen festen Wert nimmt, sondern diesen aus der Nachbarschaft eines Pixels berechnet. So ist der Algorithmus gegen verschiedene Helligkeiten im Bild robuster. Dennoch muss auch hier bestimmt werden, aus wie vielen Nachbarn das Threshold berechnet werden soll, was dazu führt, dass bei einigen Bildern das Thresholding gut funktioniert und bei anderen wiederum nicht. Außerdem wird auch hier nie konsequent alles herausgefiltert, was keine Korkpore ist (Abbildung 3.2), was das adaptive Thresholding wie auch andere traditionelle Bildverarbeitungsmethoden nicht ideal für dieses Anwendungsfeld macht, da es zu starr ist.

Abbildung 3.2: Detektion mittels Adaptivem Thresholding

3.2 Lokales Maximum/Minimum

Die Detektion lokaler Maxima und Minima ist eine grundlegende Technik in der Bildverarbeitung, die verwendet wird, um bedeutende Merkmale in einem Bild zu identifizieren. Lokale Maxima sind Punkte, an denen die Intensität eines Pixels höher ist als die seiner Nachbarn, während lokale Minima Punkte sind, an denen die Intensität eines Pixels niedriger ist als die seiner Nachbarn. Diese Merkmale sind nützlich für eine Vielzahl von Anwendungen, einschließlich Merkmalsextraktion, Bildsegmentierung und Objekterkennung.

3.2.1 Grundprinzipien der Detektion

Die Detektion von lokalen Maxima und Minima basiert auf der Analyse der Intensitätswerte der Pixel in einem Bild. Ein Pixel wird als lokales Maximum bezeichnet, wenn seine Intensität größer ist als die aller benachbarten Pixel. Umgekehrt wird ein Pixel als lokales Minimum bezeichnet, wenn seine Intensität kleiner ist als die aller benachbarten Pixel. Durch voriges Glätten des Bildes werden kleinere Intensitätsschwankungen

reduziert, wodurch nur größere, signifikante Veränderungen als lokale Extrema erkannt werden. Dies minimiert die Anzahl der False Positives, indem Rauschen unterdrückt und unbedeutende Variationen im Bild ausgeblendet werden

Mathematische Grundlagen

Die mathematische Grundlage für die Detektion lokaler Maxima und Minima kann durch die Untersuchung der Intensitätsfunktion $I(x, y)$ beschrieben werden. Ein Pixel (x, y) ist ein lokales Maximum, wenn folgende Bedingung erfüllt ist:

$$I(x, y) > I(x + i, y + j) \quad \forall (i, j) \in \{-1, 0, 1\} \setminus \{(0, 0)\} \tag{3.3}$$

Analog dazu ist ein Pixel (x, y) ein lokales Minimum, wenn:

$$I(x, y) < I(x + i, y + j) \quad \forall (i, j) \in \{-1, 0, 1\} \setminus \{(0, 0)\} \tag{3.4}$$

Diese Bedingungen werden in der Regel durch die Anwendung von Nachbarschaftsoperationen auf das Bild überprüft, wobei jeder Pixel mit seinen Nachbarn verglichen wird.

3.2.2 Ergebnisse

Das Erkennen lokaler Minima ist ein guter Ansatz, da Korkporen sich farblich eindeutig von ihrer Umgebung unterscheiden. Auch hier muss aber die Größe des Rasters aus Nachbarpixeln übergeben werden, was den Algorithmus weniger robust gegen unterschiedliche Größen macht. Außerdem werden in Bereichen, die mit vielen Korkporen gefüllt sind, die übersehen, die heller sind als andere Korkporen. Ein weiteres Problem sind dunkle Stellen, die keine Korkporen sind. So werden Punkte im braunen Streifen bei dem Golden Delicious auch zu lokalen Minima.

Abbildung 3.3: Detektion mittels lokaler Minima

3.3 Blob-Detektor

Blob-Detektion ist eine grundlegende Technik in der Bildverarbeitung, die darauf abzielt, zusammenhängende Pixel zu identifizieren, die sich in ihrer Helligkeit oder Farbe von ihrer Umgebung unterscheiden. Diese Regionen, auch als 'Blobs' bezeichnet, können unterschiedlich groß, rund oder unregelmäßig geformt sein. Die Erkennung solcher Strukturen wird in vielen Anwendungen der Computer Vision, wie Objekterkennung, Bildsegmentierung und Mustererkennung, genutzt.

3.3.1 Grundprinzipien der Blob-Detektion

Blobs sind Regionen in einem Bild, die sich in verschiedenen Merkmalen wie Helligkeit oder Farbe von den benachbarten Pixeln unterscheiden. Ziel der Blob-Erkennung ist es, diese Regionen durch Analyse der Bildmerkmale zu finden.

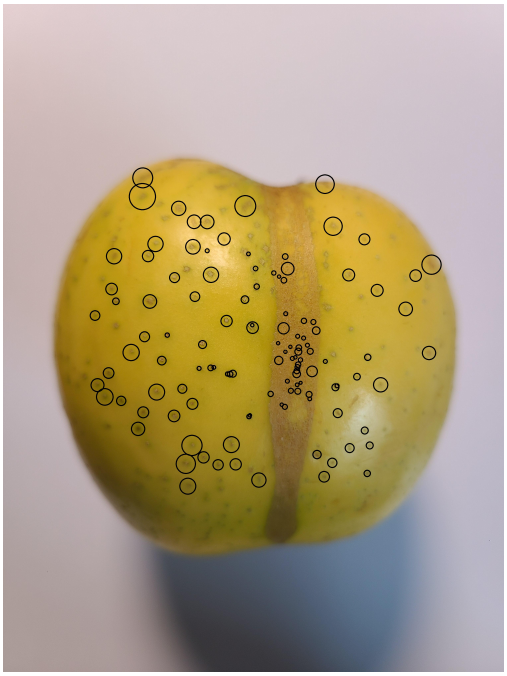
3.3.2 Ansätze zur Detektion von Blobs

Um Blobs auf einem Bild zu erkennen, kann sich verschiedener Algorithmen bedient werden. Dazu gehören unter anderem auch Thresholding oder die Detektion lokaler Extrema. Die Nutzbarkeit dieser Algorithmen unterscheidet sich dabei je nach Anwendungsfeld, wobei auch eine Kombination mehrerer Algorithmen vorteilhaft sein kann. Kaspers [2011] stellt in seiner Arbeit einige vor.

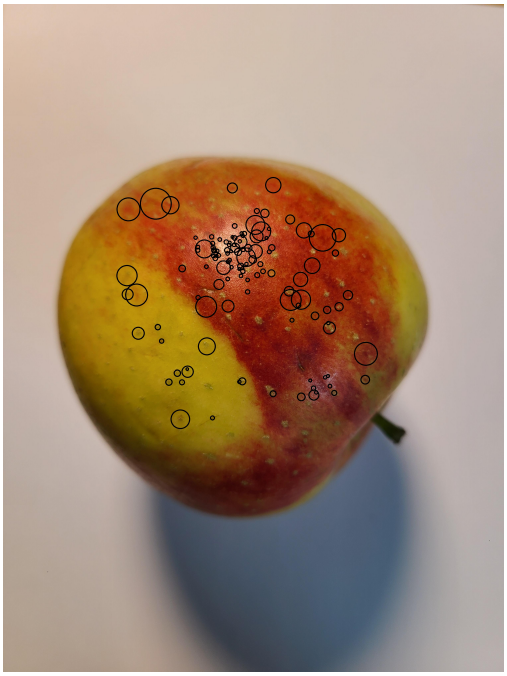
3.3.3 Ergebnisse

Grundsätzlich ist das Erkennen von zusammenhängenden Regionen durch Farbunterschiede oder Helligkeit ein guter Ansatz um Korkporen zu erkennen, da sich diese fast immer von ihrer Umgebung deutlich unterscheiden. Da jeder Apfel von Natur aus jedoch Stellen oder sonstige Merkmale hat, die in ihrer Form und Größe einer Korkpore ähnlich sein können, kann es immer zu Fehldetektionen kommen. Der Golden Delicious zum Beispiel hat in seiner Mitte einen Streifen, die auch kleine Punkte aufweisen. Diese sind in ihrer Form einer Korkpore sehr ähnlich, weshalb der Detektor hier angeschlagen hat. Ein anderes Problem kann auftreten, wenn der Apfel an einer Stelle zu stark beleuchtet wird. Dadurch entstehen genau wie bei dem Golden Delicious auf dem Streifen sehr kleine Punkte, die vom Detektor als Korkporen identifiziert werden.

Schwierig ist es also, eine korrekte Erkennung über den ganzen Apfel hinweg zu erreichen, da die Korkporen sich zum Beispiel in Farbe, Größe oder Intensität voneinander unterscheiden. Die Eingabeparameter beschränken den Detektor jedoch auf eben diese Voreinstellung, was dazu führt, dass die Varianz der Korkporen nicht abgedeckt werden kann. Wenn die Parameter zu locker sind, werden zu viele Korkporen erkannt, wie oben beschrieben, zu enge Parameter sorgen wiederum für eine geringe Erkennungsrate. Dies wäre zumindest in Teilen zu beheben, indem mehrere Blob-Detektoren nacheinander dasselbe Bild verarbeiten. Jedoch können durch die Unterschiede der Korkporen oder der Beleuchtung nicht alle Bereiche und damit alle Korkporen identifiziert werden. Dazu kommt, dass verschiedene Sorten wieder verschiedene Parameter brauchen.



(a) In dem braunen Strich wurden fälschlicherweise Korkporen erkannt, da auch hier kleine Punkte auftauchen die sich von ihrer Umgebung unterscheiden.



(b) Beim Wellant werden fälschlicherweise an der beleuchteten Stelle viele kleine Spots erkannt.

Abbildung 3.4: Detektion mittels Blob-Detektor

3.4 Fazit

Die in diesem Abschnitt beschriebenen Ansätze können weiter optimiert werden, so dass die Erkennung besser funktioniert, aber die Probleme bleiben bestehen. Die Oberfläche eines Apfels ist zu vielfältig, um einen Algorithmus zu konstruieren, der immer eine hohe Erkennungsrate erzielt. Die Ergebnisse machen deutlich, dass die Erkennung von Korkporen mit herkömmlichen Methoden nicht zuverlässig möglich ist. Unterschiedliche Formen, Farben, Beleuchtungen und andere Merkmale auf den Äpfeln erfordern unterschiedliche Algorithmen und Parameter. So kann ein bestimmter Detektor oder Filter für einen Apfel eine gute Erkennungsrate erzielen, für den nächsten sind jedoch neue Parameter erforderlich. Aus diesem Grund werden im nächsten Abschnitt der Arbeit neuronale Netze verwendet und verglichen, da diese aufgrund ihres eigenen Ansatzes wesentlich robuster gegenüber Vielseitigkeit sind als die traditionellen Methoden.

4 Neuronale Netze

In den letzten Jahrzehnten ist die Erkennung von Objekten in Bildern durch neuronale Netze effizienter geworden. Früher basierten die Erkennungsmethoden auf manuell erstellten Merkmalen oder einfachen Klassifikatoren, die jedoch aufgrund der Komplexität der Bilder an ihre Grenzen stießen. Mit der Entwicklung von neuronalen Netzen, insbesondere Convolutional Neural Network (CNN)s, wurde die Effizienz und Genauigkeit der Objekterkennung in Bildern erheblich verbessert.

Krizhevsky, Sutskever und Hinton [2012] war einer der ersten, der CNNs zur Bildklassifikation auf dem ImageNet-Datensatz einsetzte und damit deutlich bessere Ergebnisse als mit klassischen Methoden erzielte. Die CNNs verwenden mehrere Schichten von Faltungsoperationen und Pooling-Operationen, um Merkmale aus Bildern zu extrahieren und anschließend zu klassifizieren. Seitdem haben sich CNNs als dominierende Technologie in der Bildverarbeitung etabliert und bilden die Grundlage für viele neue Modelle zur Objekterkennung.

R-CNNs und später YOLO oder SSD brachten weitere Fortschritte. Das Region-based Convolutional Neural Network (R-CNN) von Girshick, Donahue, Darrell und Malik [2014] führte das Konzept der regionsbasierten Erkennung ein. Dabei werden zunächst potentielle Objekte im Bild erkannt und anschließend klassifiziert. You Only Look Once (YOLO) von Redmon, Divvala, Girshick und Farhadi [2016] und Single Shot Multibox Detector (SSD) von Liu, Anguelov, Erhan, Szegedy, Reed, Fu und Berg [2016] hingegen fassen diese beiden Schritte in einem zusammen.

4.1 Modellwahl

In dieser Arbeit wurden die Modelle YOLO, SSD, Faster R-CNN, RetinaNet und U-Net ausgewählt, um eine umfassende Analyse und einen Vergleich der Leistungsfähigkeit moderner Objekterkennungs- und Segmentierungsmodelle zu ermöglichen. Diese Auswahl

basiert auf der breiten Anwendung und den spezifischen Anforderungen der Aufgabenstellung.

Die gewählten Modelle decken ein breites Spektrum an Ansätzen zur Objekterkennung und Bildsegmentierung ab, von One-Staged Modellen (YOLO, SSD, RetinaNet) bis hin zu Two-Staged Modellen (Faster R-CNN) und speziellen Segmentierungsmodellen (U-Net). Der Unterschied dieser drei Arten und die Funktionsweise der einzelnen Modelle wird im Folgenden genauer erläutert. So wird ersichtlich, weshalb ein Vergleich verschiedener Modelle dieser Arten wichtig ist. Desweiteren ist das Verständnis dieser Unterscheidung relevant für die spätere Einordnung der Ergebnisse.

4.1.1 One-Staged Modelle (YOLO, SSD, RetinaNet)

One-Staged Modelle, auch bekannt als einstufige Detektionsmodelle, führen die Aufgaben der Objekterkennung und Klassifizierung in einem Durchlauf durch. Diese Modelle unterteilen das Bild in ein Raster und führen für jedes Rasterfeld gleichzeitig eine Objekterkennung und Klassifizierung durch.

Die grundsätzliche Funktionsweise eines One-Staged Modells kann wie folgt beschrieben werden:

- **Rasterung des Bildes:** Das Eingabebild wird in ein Raster von festen Größen unterteilt. Jedes Rasterfeld wird individuell untersucht.
- **Feature Extraction:** Durch die Anwendung von Convolutional Layers werden Merkmale aus dem Bild extrahiert.
- **Bounding Box und Klassifizierung:** Für jedes Rasterfeld wird eine Bounding Box vorhergesagt, die das Objekt umgibt, und eine Wahrscheinlichkeitsverteilung über mögliche Objektklassen berechnet.

YOLO

Architektur Die Architektur von You Only Look Once (YOLO) basiert auf einem einzigen CNN, das das Eingabebild in ein Gitter unterteilt und für jede Gitterzelle mehrere Bounding-Box-Vorhersagen und deren Wahrscheinlichkeiten erzeugt. Die Hauptkomponenten von YOLO sind:

- **Backbone-Netzwerk:** Ein tiefes CNN dient als Feature-Extractor. Dieses Netzwerk extrahiert relevante Merkmale aus dem Eingabebild und liefert eine Feature Map.
- **Gitterunterteilung:** Das Eingabebild wird in ein $S \times S$ Gitter unterteilt. Jede Gitterzelle ist verantwortlich für die Vorhersage von B Bounding-Boxen und deren Konfidenzwerten sowie den Wahrscheinlichkeiten für C Klassen.
- **Bounding-Box-Vorhersage:** Jede Gitterzelle liefert mehrere Bounding-Box-Vorhersagen, die jeweils eine Position, Größe, und Konfidenzwert enthalten. Der Konfidenzwert gibt an, wie sicher das Modell ist, dass die Box ein Objekt enthält und wie präzise die Vorhersage ist.
- **Klassifikation:** Zusätzlich zu den Bounding-Boxen liefert jede Gitterzelle Wahrscheinlichkeiten für die Anwesenheit bestimmter Klassen innerhalb der Zelle.

SSD

Architektur Die Architektur von Single Shot Multibox Detector (SSD) basiert auf einem CNN, das verwendet wird, um Merkmale aus dem Eingabebild zu extrahieren. SSD erweitert ein Standard-CNN durch die Hinzufügung von mehreren Convolutional Layers am Ende des Netzwerks, die es ermöglichen, Objekte in verschiedenen Größen und auf verschiedenen Auflösungsebenen zu erkennen. Diese werden im folgenden n.her beschrieben.

- **Multiskalige Feature Maps:** SSD verwendet mehrere Feature Maps, die aus den verschiedenen Ebenen des Netzwerks stammen, um Objekte auf verschiedenen Skalen zu erkennen. Jede dieser Feature Maps ist verantwortlich für das Erkennen von Objekten in einer bestimmten Größenskala, was die Fähigkeit des Modells verbessert, sowohl kleine als auch große Objekte zu detektieren.
- **Default Boxes:** SSD verwendet eine Technik namens Default Boxes, die festgelegte Ankerboxen mit verschiedenen Seitenverhältnissen und Skalen auf jeder Feature Map sind. Für jeden Rasterpunkt der Feature Map wird eine Reihe von Default Boxes erzeugt, die dann mit den Ground Truth Boxen verglichen werden, um Objekte zu erkennen. Diese Default Boxes werden während des Trainings mit den tatsächlichen Objektboxen verglichen und angepasst, um die Position und Größe der erkannten Objekte besser abzugleichen.

- **Vorhersagen:** Bei einem SSD wird die Vorhersage in eine Klassifikation und eine Positionsregression unterteilt. Um zu bestimmen, ob eine Default Box ein Objekt enthält und wenn ja, welches, wird sie klassifiziert. Desweiteren wird eine Positionsregression durchgeführt, um die genaue Position und Größe der Box zu erkennen. Beide Teile werden separat auf den Feature Maps angewandt.

RetinaNet

RetinaNet wurde von Lin, Goyal, Girshick, He und Dollár [2017] entwickelt und ist für seine Fähigkeit bekannt ist, eine hohe Erkennungsgenauigkeit mit effizienter Berechnung zu kombinieren. RetinaNet nutzt das Konzept des Focal Loss, um das Ungleichgewicht zwischen Hintergrund- und Vordergrundklassen zu bewältigen, was es besonders effektiv für die Erkennung kleiner Objekte in großen Bildern macht.

Architektur Die Architektur von RetinaNet besteht aus folgenden Hauptkomponenten:

- **Backbone-Netzwerk:** Das Backbone-Netzwerk ist ein tiefes CNN das verwendet wird, um Merkmale aus dem Eingabebild zu extrahieren. Diese Merkmale werden dann durch ein Feature Pyramid Network (FPN) weiterverarbeitet, um Feature Maps auf verschiedenen Auflösungsebenen zu erzeugen.
- **Feature Pyramid Network (FPN):** Das FPN erzeugt eine pyramidenartige Struktur von Feature Maps, die es dem Modell ermöglicht, Objekte in verschiedenen Größen zu erkennen. Diese Pyramide stellt sicher, dass sowohl große als auch kleine Objekte effektiv erkannt werden können.
- **Klassifikations- und Regressionsköpfe:** Zwei getrennte Köpfe werden auf jeder Ebene der Pyramide angewendet: ein Klassifikationskopf, der die Wahrscheinlichkeit jedes Ankers einer Klasse zuordnet, und ein Regressionskopf, der die genauen Grenzen der Objekte vorhersagt.

Focal Loss Das Konzept des Focal Loss wurde entwickelt, um das Ungleichgewicht zwischen einfachen erkennbaren und schwierig erkennbaren Objekten in der Objekterkennung zu bewältigen. Ohne Focal Loss kann es passieren, dass die Objekte, die grundsätzlich leichter zu erkennen sind, auch häufiger vorhergesagt werden, als die schwierigen

Objekte. Der Focal Loss setzt dem entgegen, indem er den Verlust für leicht erkennbaren Objekte reduziert und den Fokus auf schwer erkennbaren Objekten erhöht. Dies ermöglicht eine bessere Balance und führt zu einer höheren Erkennungsgenauigkeit, insbesondere bei schwierigen Objekten. Durch diese Anpassung wird das Training effizienter und das Modell lernt besser, schwierige Objekte zu erkennen.

4.1.2 Two-Staged Modelle (R-CNN)

Two-Staged Modelle, oder zweistufige Detektionsmodelle, trennen die Objekterkennung und Klassifizierung in zwei separate Schritte.

- **Region Proposal:** Zunächst werden mögliche Regionen im Bild identifiziert, die Objekte enthalten könnten. Dies geschieht oft durch Algorithmen wie Selective Search oder Region Proposal Networks (RPN).
- **Klassifizierung:** In einem zweiten Schritt werden die vorgeschlagenen Regionen klassifiziert und die präzisen Bounding Boxes vorhergesagt.

Die Funktionsweise eines Two-Staged Modells lässt sich wie folgt detaillierter beschreiben:

- **Region Proposal Networks (RPN):** Diese Netzwerke scannen das Bild und erzeugen Vorschläge für Regionen, die Objekte enthalten könnten.
- **RoI Pooling:** Die vorgeschlagenen Regionen werden auf eine einheitliche Größe gebracht und durch ein CNN geleitet.
- **Klassifikation und Bounding Box Regression:** Jede Region wird klassifiziert und die genaue Position der Bounding Box wird berechnet.

Faster R-CNN

Faster R-CNN verbessert die vorherigen R-CNN-Modelle (R-CNN und Fast R-CNN) durch die Einführung eines RPN, das die Erzeugung von Vorschlagsregionen direkt in das Netzwerk integriert und so die Geschwindigkeit der Objekterkennung erheblich erhöht.

Architektur Die Architektur von Faster R-CNN besteht aus drei Hauptkomponenten:

- **Feature Extractor:** Ein CNN wird verwendet, um Merkmale aus dem Eingabebild zu extrahieren. Diese Merkmale werden in Form von Feature Maps als Eingabe sowohl für das Region Proposal Network als auch für den Klassifikations- und Regressionskopf verwendet.
- **Region Proposal Networks (RPN):** Das RPN ist ein kleines Netzwerk, das auf den Feature Maps operiert und Vorschlagsregionen generiert. Es gleitet mit einem kleinen Fenster über die Feature Maps und erzeugt für jedes Fenster eine Reihe von Ankern unterschiedlicher Größen und Seitenverhältnisse. Diese Anker werden dann daraufhin überprüft, ob sie ein Objekt enthalten oder nicht und dann dementsprechend klassifiziert und ihre Positionen regressiert.
- **RoI Pooling und Klassifikation:** Die durch das RPN erzeugten Region Proposals werden mittels Regions of Interest Pooling in eine feste Größe transformiert, die als Eingabe für die nachfolgenden Fully Connected Layer dient. Diese Schichten führen die finale Klassifikation und die präzisere Positionsregression der Vorschlagsregionen durch.

4.1.3 Segmentierungsmodelle (U-Net)

U-Net

Mit U-Net von Ronneberger, Fischer und Brox [2015] wird in die Auswahl noch ein Segmentierungsmodell aufgenommen, das sich dadurch auszeichnet, dass es deutlich schneller als die üblichen Segmentierungsmodelle funktioniert und darüber hinaus mit nur sehr wenigen Trainingstag auskommt.

Architektur Die U-Net-Architektur zeichnet sich durch ihre symmetrische U-Form aus, die aus einem kontraktiven Pfad und einem expansiven Pfad besteht. Diese Struktur ermöglicht es, sowohl lokale als auch globale Informationen zu nutzen, um präzise Segmentierungen zu erzeugen.

- **Kontraktiver Pfad:** Der kontraktive Pfad folgt der typischen Architektur eines CNN. Er besteht aus mehreren Schichten von 3x3-Konvolutionen, gefolgt von einer ReLU-Aktivierungsfunktion und 2x2-Max-Pooling-Schichten zur Reduzierung der Dimensionen. Jede Downsampling-Stufe verdoppelt die Anzahl der Feature-Kanäle.
- **Bottleneck:** Der Bottleneck-Bereich verbindet den kontraktiven und den expansiven Pfad und besteht aus mehreren Convolutional Layers, die die tiefsten Merkmale des Eingabebildes erfassen.
- **Expansiver Pfad:** Der expansive Pfad besteht aus einer Folge von 2x2-Upkonvolutionen, die die räumliche Auflösung der Merkmalskarten erhöhen, gefolgt von einer 3x3-Konvolution und einer ReLU-Aktivierung. In jeder Stufe werden die Merkmale mit den entsprechenden Merkmalskarten aus dem kontraktiven Pfad über sogenannte Skip-Verbindungen kombiniert. Diese Verbindungen helfen, die Lokalisierungsgenauigkeit zu verbessern, indem sie Kontextinformationen aus höheren Auflösungen bereitstellen.
- **Endschicht:** Die letzte Schicht des U-Net besteht aus einer 1x1-Konvolution, die die Anzahl der Ausgabekanäle auf die Anzahl der Klassen reduziert, die segmentiert werden sollen.

4.1.4 Fazit

Die Auswahl von YOLO, SSD, Faster R-CNN und RetinaNet ermöglicht es, die Leistungsfähigkeit von One-Staged und Two-Staged Modellen zu vergleichen. One-Staged Modelle wie YOLO (Redmon, Divvala, Girshick und Farhadi [2016]) und SSD (Liu, Anguelov, Erhan, Szegedy, Reed, Fu und Berg [2016]) sind bekannt für ihre hohe Verarbeitungsgeschwindigkeit und Effizienz, was sie besonders geeignet für Echtzeit-Anwendungen macht. RetinaNet von Lin, Goyal, Girshick, He und Dollár [2017], ein weiteres One-Staged Modell, bietet durch die Einführung der Focal Loss-Funktion eine verbesserte Erkennungsgenauigkeit, insbesondere bei schwer zu erkennenden Objekten.

Two-Staged Modelle wie Faster R-CNN (Ren, He, Girshick und Sun [2015]) bieten hingegen eine höhere Erkennungsgenauigkeit durch die getrennte Phase der Region Proposal und Klassifikation. Diese Modelle sind laut Huang, Rathod, Sun, Zhu, Korattikara, Fathi, Fischer, Wojna, Song, Guadarrama u. a. [2017] besonders geeignet für Anwendungen, die

höchste Präzision erfordern, wie z.B. die medizinische Bildverarbeitung. Durch den Vergleich dieser Modelle können die Vor- und Nachteile der verschiedenen Ansätze umfassend analysiert werden.

Die Einbeziehung von U-Net in diese Arbeit ermöglicht es, die Fähigkeiten der Bildsegmentierung zu evaluieren. U-Net von Ronneberger, Fischer und Brox [2015] ist speziell für die präzise Segmentierung von Bildern entwickelt worden und hat sich in der medizinischen Bildverarbeitung als äußerst effektiv erwiesen.

Zusammengefasst ermöglicht die Auswahl dieser fünf Modelle eine fundierte und umfassende Analyse, bei der die Genauigkeit der Objekterkennung und Bildsegmentierung berücksichtigt wird. Dies stellt sicher, dass die besten Modelle für spezifische Anwendungsfelder identifiziert und die Ergebnisse dieser Arbeit auf eine breite Palette von Anwendungen übertragen werden können.

4.2 Versuchsaufbau

Die Modelle werden jeweils mit dem Datensatz trainiert und anschließend anhand einiger Testbilder evaluiert. Um einen fairen Vergleich zu ermöglichen, trainieren alle Modelle mit den gleichen Voreinstellungen. So wird sichergestellt, dass kein Modell z.B. durch längeres Training bessere Ergebnisse liefert.

Jedes Modell trainiert mit 200 Epochen. Um gute Ergebnisse zu erzielen, ist es wichtig, die optimale Anzahl von Epochen zu wählen. Dabei ist zu beachten, dass zu wenige Epochen die Ergebnisse verschlechtern können, da das Training zu früh beendet wird, obwohl das Modell noch Lernfortschritte macht. Auf der anderen Seite führen zu viele Epochen zu einem Overfitting, bei dem das Modell den bestmöglichen Lerneffekt aus den Daten ziehen konnte und durch weiteres Training falsche Informationen aus dem Datensatz extrahiert. In diesem Fall wurden 200 Epochen gewählt, da die Modelle nach 200 Epochen zwar immer noch besser werden können, aber nicht mehr signifikant.

Die in dieser Arbeit gewählten Trainingseinstellungen liefern nicht die bestmöglichen Ergebnisse für die Modelle. Da es aber darum geht, die Modelle miteinander zu vergleichen, ist dies vernachlässigbar, da ein Vergleich genau die gleichen Ergebnisse liefern würde, wenn die bestmöglichen Voreinstellungen gewählt worden wären. Wichtig ist nur, dass alle Modelle mit den gleichen Einstellungen verglichen werden. Die optimalen Parameter für das Training könnten dann in einem möglichen Nachbearbeitungsschritt mit dem

besten Modell ermittelt werden. Dies ist jedoch nicht Gegenstand dieser Arbeit. Nachbearbeitungsschritt

4.3 Datensatz

Für den Trainingsdatensatz wurden 121 Bilder mit einem Samsung Galaxy S20+ aufgenommen. Diese Bilder haben eine Auflösung von 3024x4032 Pixel. Insgesamt wurden zwei Äpfel aus verschiedenen Positionen fotografiert, um eine möglichst vielfältige Ansammlung von Korkporen zu erhalten. So haben die Korkporen unterschiedliche Größen, Anordnungen, Farben und Umgebungen. Dadurch soll ein möglichst robustes Modell trainiert werden. Da in diesem Training nur eine Klasse zu identifizieren ist, reichen 121 Bilder aus, um gute Ergebnisse zu erzielen. Bei einer größeren Anzahl von Klassen würden 121 Bilder nicht ausreichen, damit das Modell eine Klasse sicher vorhersagen kann. In der Arbeit von Shahinfar, Meek und Falzon [2020] wird deutlich, dass die Anzahl der Bilder logarithmisch mit der Leistung des Trainings korreliert. Der stärkste Anstieg ist bei einer Anzahl von weniger als 100 Bildern zu verzeichnen. Danach steigt die Leistung zwar kontinuierlich, aber nicht mehr so stark. 121 Bilder sind also nicht optimal, aber ausreichend, um die Modelle effektiv zu trainieren. Die Beleuchtung und der Hintergrund der Bilder sind ähnlich, was zu Overfitting führen kann, da die Modelle trainiert werden, dass diese Umstände relevant sind. Zusätzlich wurden zwei Apfelsorten (Golden Delicious und Wellant) fotografiert. Bei diesen beiden Sorten sind die Korkporen deutlich sichtbar. Probleme könnten später bei anderen Sorten auftreten, deren Korkporen möglicherweise andere Merkmale aufweisen.

Zusammenfassend kann gesagt werden, dass die Bilder einen ausreichend guten Datensatz liefern, indem sie eindeutig zeigen, was eine Korkpore ist. Es ist jedoch darauf hinzuweisen, dass der Datensatz ziemlich homogen ist, was dazu führen kann, dass die Modelle nicht sehr robust gegenüber Änderungen im Bildaufbau sind. Spätere Tests werden dieses Problem näher beleuchten.

4.4 Labelisierung

Die Labels werden als Bounding Boxes gespeichert und für jedes Modell in das richtige Format transformiert. Dadurch wird sichergestellt, dass alle Modelle mit den exakt

Abbildung 4.1: Auswahl an Bildern aus dem Datensatz

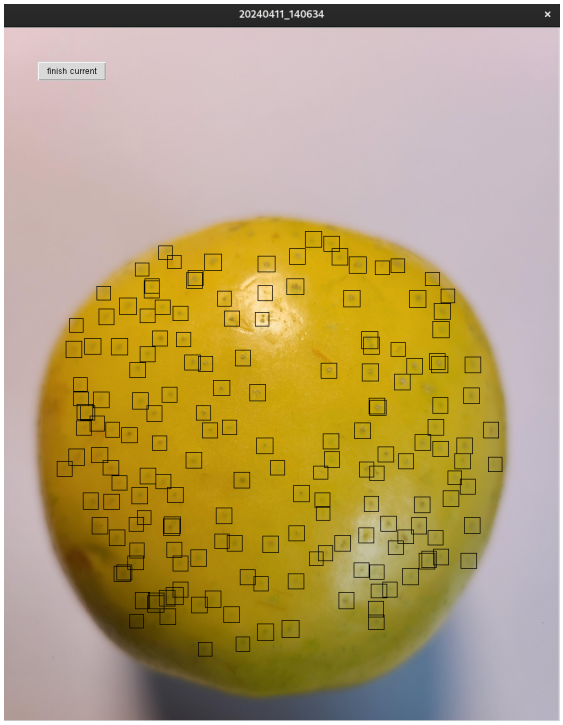


Abbildung 4.2: Interface des Tools zum Erstellen der Labels

gleichen Positionen der Labels arbeiten. Um die Korkporen als solche zu beschriften, wurde ein spezielles Programm in Python geschrieben. Für jedes angeklickte Feld wird die Klasse C, ein Tupel aus (x, y) für den Mittelpunkt der Bounding Box und ein Tupel aus (w, h) für die Breite und Höhe gespeichert. Wenn alle Korkporen markiert sind, wird das Bild als beschriftet gekennzeichnet. Für YOLO werden diese Bounding Boxes direkt zum Trainieren verwendet, für SSD, Faster R-CNN und RetinaNet wurde ein Programm geschrieben, das die Bounding Boxes in Pascal VOC formatiert. Für U-Net wurde ein Programm geschrieben, das die Bounding Boxes in ein JSON-Format umwandelt. Wenn der Datensatz vergrößert wird, kann auf diese Weise sichergestellt werden, dass jedes Modell den Datensatz im richtigen Format zur Verfügung hat. Dadurch werden Fehler bei der Aufbereitung vermieden.

4.5 Vorbereitung und Arten von Testbildern

Um die Modelle zu testen, wurden neue Bilder erstellt. Da die Modelle miteinander verglichen werden, ist es wichtig, dass alle mit den gleichen Bildern getestet werden. Dabei

spielt es keine Rolle, ob die Bilder den Trainingsbildern ähneln. Die Testbilder wurden mit einem IPhone SE der ersten Generation aufgenommen und sollen verschiedene Aspekte abdecken, um eine möglichst große Auswahl an Testszenarien zu schaffen.

Die neu aufgenommenen Bilder bestehen aus Äpfeln einer Sorte, mit der die Modelle nicht trainiert wurden. Dazu gehören Crimson Snow, Elstar, Granny Smith und Red Jonaprince. Außerdem wurde ein Wellant mit einer fehlerhaften Stelle fotografiert, um zu vergleichen, wie die Modelle mit solchen Stellen umgehen. Ansonsten wurden Bilder mit verschiedenen Helligkeitsstufen aufgenommen und ein Wellant wurde von oben und unten fotografiert. Außerdem wurden Äpfel vor einem Hintergrund fotografiert. In einem Bild wird der Apfel in der Hand gehalten, in einem anderen liegt er in einer Schale, umgeben von anderen Objekten, die als Ablenkung dienen sollen.

Darüber hinaus kann die Robustheit der Modelle anhand von Faktoren überprüft werden, die durch Bildverarbeitung simuliert werden können. So sind Modelle, die auch bei schlechter Bildqualität eine hohe Erkennungsrate aufweisen, solchen vorzuziehen, die nur bei Bildern, die den Trainingsbildern ähnlich sind, eine gute Erkennung liefern. Durch gezielte Tests mit verrauschten oder herunterskalierten Bildern können die Grenzen und Stärken der Modelle überprüft werden.

In dieser Arbeit wird daher die Leistungsfähigkeit der Modelle u.a. unter Einbeziehung von verarbeiteten Bildern analysiert. Dadurch wird nicht nur die Robustheit und Adaptivität der Modelle getestet, sondern es können auch mögliche Vorverarbeitungsschritte identifiziert werden, die später im Anwendungsfeld zu einer verbesserten Performance führen.

Grundsätzlich gibt es viele Bildverarbeitungsschritte in der digitalen Bildverarbeitung, die eingesetzt werden, um die Qualität und Nutzbarkeit von Bildern oder Videos zu verbessern. Dieses Kapitel konzentriert sich auf die Techniken der Rauschunterdrückung bzw. Rauschaddition und der Bildreduktion. Diese beiden Techniken werden im Detail erläutert, wobei beim Rauschen insbesondere auf das Gaussche Rauschen und das Salz-und-Pfeffer-Rauschen eingegangen wird, da diese nach Boncelet [2009] die am häufigsten auftretenden Varianten sind.

4.5.1 Rauschen

Gaussches Rauschen

Das Gaussche Rauschen, auch bekannt als additiv weißes Rauschen, tritt häufig in Bildern auf und wird durch eine Normalverteilung der Pixelwerte beschrieben. Es führt zu zufälligen Schwankungen um den Mittelwert der Pixelintensität und kann durch verschiedene physikalische Prozesse entstehen, wie z.B. thermische Rauschprozesse in den Sensoren.

Zunächst werden der Mittelwert μ und die Standardabweichung σ des Gausschen Rauschens bestimmt. Für jedes Pixel wird ein Rauschwert N aus einer Normalverteilung $\mathcal{N}(\mu, \sigma^2)$ generiert und zu dem ursprünglichen Pixelwert $I(x, y)$ hinzugefügt, um den neuen Pixelwert $I'(x, y)$ zu erhalten. Falls $I'(x, y)$ außerhalb des zulässigen Wertebereichs liegt, wird er entsprechend begrenzt.

Mathematische Darstellung Die Umsetzung kann mathematisch wie folgt beschrieben werden:

$$I'(x, y) = I(x, y) + \mathcal{N}(\mu, \sigma^2) \tag{4.1}$$

Dabei stellt $\mathcal{N}(\mu, \sigma^2)$ eine Zufallsvariable dar, die normalverteilt ist mit dem Mittelwert μ und der Varianz σ^2.

Auswirkungen und Anwendungen Gaußsches Rauschen kann die Bildqualität erheblich beeinträchtigen. Neuronale Netze, die mit Gaußschem Rauschen umgehen können, sind in der Praxis zuverlässiger und können auch bei verrauschten Eingangsdaten gute Ergebnisse liefern.

Mit Hilfe des beschriebenen Ansatzes wurden Testbilder erzeugt, die später sicherstellen, dass die Modelle auch in realen Anwendungsfällen robust gegenüber Gaußschem Rauschen sind.

(a) Normales Bild (b) Standardabweichung von 0.5

Abbildung 4.3: Vergleich der Bilder mit Gausschem Rauschen

Salz-und-Pfeffer-Rauschen

Salz- und Pfeffer-Rauschen wird häufig durch Datenübertragungsfehler, Störungen während der Bildaufnahme oder fehlerhafte Speicheroperationen verursacht. Es ist in Bildern besonders auffällig, da es starke Kontraste zwischen den Rauschpixeln und den umgebenden Pixeln erzeugt. Diese Art von Rauschen kann die Bildqualität erheblich beeinträchtigen, da wichtige Bilddetails verdeckt oder falsch dargestellt werden. Der Name 'Salz und Pfeffer' kommt daher, dass die hellen Rauschpixel als 'Salz' (weiß) und die dunklen Rauschpixel als 'Pfeffer' (schwarz) bezeichnet werden.

Die Bestimmung der Rauschdichte ρ legt fest, welcher Anteil der Pixel von Salz-und-Pfeffer-Rauschen betroffen ist; durch die Erzeugung von Zufallszahlen für jedes Pixel und den Vergleich dieser mit ρ wird entschieden, ob ein Pixel unverändert bleibt, zu 'Pfeffer' (schwarz) oder zu 'Salz' (weiß) wird.

Mathematische Darstellung Die Umsetzung kann mathematisch wie folgt beschrieben werden:

(a) Normales Bild

(b) Rauschdichte von 0.25

Abbildung 4.4: Vergleich der Bilder mit Salz-und-Pfeffer-Rauschen

- Für jedes Pixel $I(x, y)$ im Originalbild:

$$I'(x,y) = \begin{cases} 0 & \text{wenn } r \leq \frac{\rho}{2} \\ 255 & \text{wenn } \frac{\rho}{2} < r \leq \rho \\ I(x,y) & \text{wenn } r > \rho \end{cases} \tag{4.2}$$

Es ist wichtig sicherzustellen, dass die neuen Pixelwerte innerhalb des zulässigen Wertebereichs bleiben. Für ein 8-Bit-Graustufenbild bedeutet dies, dass die Pixelwerte entweder 0 (schwarz) oder 255 (weiß) sind.

Auswirkungen und Anwendungen Salz-und-Pfeffer-Rauschen kann signifikante Auswirkungen auf die Bildqualität haben. In der Bildverarbeitung wird dieses Rauschen oft künstlich hinzugefügt, um die Robustheit und Leistungsfähigkeit von Bildverarbeitungsalgorithmen zu testen. Durch das Hinzufügen von Salz-und-Pfeffer-Rauschen zu Testbildern kann diese Arbeit sicherstellen, dass die angewandten Modelle robust gegenüber Störungen oder Fehlern in der realen Anwendung sind.

4.5.2 Runterskalierung (Downsampling)

Herunterskalierung, auch Downsampling genannt, ist ein wichtiger Prozess in der Bildverarbeitung, bei dem die Auflösung eines Bildes reduziert wird. Das bedeutet, dass die Anzahl der Pixel, die das Bild darstellen, verringert wird, wodurch die Bildgröße und die benötigte Speicherkapazität reduziert werden. Allerdings gehen dabei auch Daten in Form von Pixelwerten verloren.

Algorithmische Umsetzung der Runterskalierung

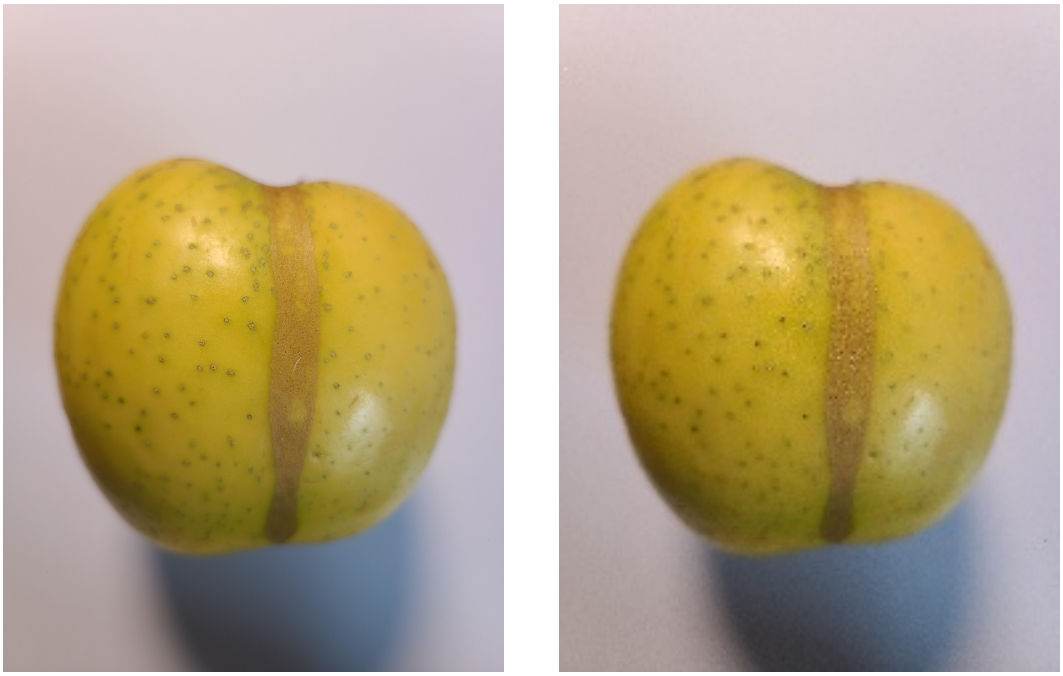
Es gibt mehrere Methoden, wie zum Beispiel Nearest Neighbor Interpolation, Bilineare Interpolation, Bikubische Interpolation oder Area-based Interpolation, um ein Bild herunterzuskalieren. Um die Runterskalierung zu näher zu erklären, wird die bikubische Interpolation betrachtet, die laut Han [2013], Patel und Mistree [2013] und Fadnavis [2014] eine der herkömmlichsten und besten Methoden ist und dazu in vielen Bildverarbeitungsprogrammen benutzt wird.

1. **Koordinatentransformation**: Zunächst werden die Koordinaten jedes Pixels im skalierten Bild auf das Koordinatensystem des Originalbildes zurückgeführt. Dies erfolgt durch die Skalierungsfaktoren für die horizontale und vertikale Achse, S_x und S_y, die aus dem Verhältnis der Abmessungen des Originalbildes zu denen des skalierten Bildes berechnet werden.

2. **Bestimmung der Nachbarpixel**: Für jedes Pixel im skalierten Bild werden die 16 nächstgelegenen Pixel im Originalbild identifiziert. Diese Nachbarpixel bilden ein 4x4-Feld, um den Zielpixelwert zu berechnen.

3. **Interpolation**: Die Pixelwerte der Nachbarpixel werden unter Berücksichtigung ihrer Entfernungen zu den Zielkoordinaten interpoliert. Dies erfolgt mittels einer kubischen Funktion, die für jeden der vier Dimensionen (horizontal, vertikal und beide diagonalen) die Gewichte berechnet.

4. **Wertzuweisung**: Der berechnete Wert wird dem entsprechenden Pixel im skalierten Bild zugewiesen.



(a) Normales Bild (b) Runterskalierung auf $\frac{1}{8}$

Abbildung 4.5: Vergleich der Bilder nach Anwendung von Downsampling

Anwendungen und Bedeutung der Runterskalierung

Die Verkleinerung ist ein wichtiger Prozess in der Bildverarbeitung, der eine effiziente Reduzierung der Bildauflösung ermöglicht. Durch die Anwendung verschiedener Interpolationsmethoden können Bilder verkleinert werden, wobei wichtige Details und Strukturen so weit wie möglich erhalten bleiben. Dies ist besonders wichtig für Anwendungen mit begrenzten Speicher- und Verarbeitungskapazitäten. Daher ist das Testen auf Bildern mit niedriger Auflösung ein wichtiger Aspekt, um die Qualität der Modelle zu überprüfen.

(a) Crimson Snow

(b) Elstar

(c) Granny Smith

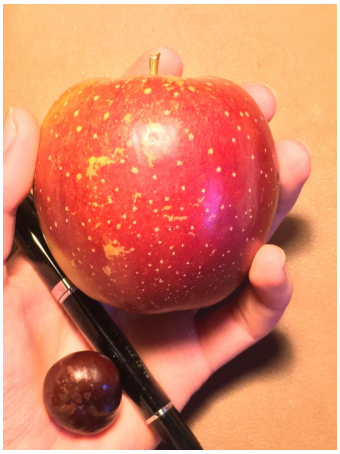
(d) Downsampled

(e) Gaussches Rauschen

(f) Salz-und-Pfeffer

(g) Helle Beleuchtung

(h) Untere Ansicht



(i) Apfel mit Hintergrund

Abbildung 4.6: Auswahl an Bildern aus dem Testdatensatz

5 Ergebnisanalyse

Um eine valide Evaluierung der Modelle zu gewährleisten, ist es unerlässlich, dass alle Modelle mit den gleichen Bildern getestet werden. Das Verhältnis der Testbilder zu den Bildern des Trainingsdatensatzes ist dabei unerheblich. Daher wurden für die Tests sowohl neue Bilder erstellt als auch Bilder verwendet, die in ihrer Struktur den Trainingsbildern sehr ähnlich sind.

Der Vergleich der Modelle erfolgt zunächst anhand der Trainingsleistung. In diesem Kontext sind die Faktoren Mean Average Precision (mAP) und der Verlust von entscheidender Bedeutung. Anhand dieser Faktoren lässt sich der Erfolg des Trainings mit den Bildern gut beurteilen.

Des Weiteren ist ein Vergleich der Modelle hinsichtlich ihrer Robustheit vorgesehen. Zu diesem Zweck wurden verschiedene Kategorien definiert, welche eine alltägliche Umgebung simulieren sollen. Des Weiteren wurden für die Modelle unbekannte Sorten in den Testdatensatz aufgenommen, d. h. Apfelsorten, mit denen die Modelle nicht trainiert wurden. Des Weiteren ist von essentieller Bedeutung, dass die Modelle eine hohe Robustheit gegenüber unterschiedlichen Bildqualitäten aufweisen. Des Weiteren wurden Bilder mit Rauschen oder Downsampling verwendet. Schließlich wurden Bilder in den Testdatensatz aufgenommen, die sich in Bezug auf den Bildaufbau von den Trainingsbildern unterscheiden. Die Bilder weisen demnach unterschiedliche Beleuchtungen, Hintergründe, Positionen sowie verfaulte oder gebrochene Stellen auf. Dies ist insbesondere relevant, da im späteren Anwendungsfeld diese Faktoren häufig vorkommen werden und ein Modell, das auch unter solchen Umständen eine hohe Erkennungsrate aufweist, wesentlich besser als andere sein kann.

Die Schaffung einer möglichst heterogenen Testumgebung dient der Evaluierung der Modelle in Bezug auf verschiedene Kategorien. Dadurch sollen die Stärken und Schwächen der Modelle in unterschiedlichen Anwendungsbereichen aufgezeigt werden, um eine fundierte Entscheidungsgrundlage für die Auswahl geeigneter Modelle zu bieten.

5.1 Statistik

In diesem Abschnitt werden die statistischen Metriken beschrieben, die zur Bewertung der Objekt-Erkennungsmodelle verwendet werden. Die Leistungsbewertung eines Modells spielt eine zentrale Rolle, einen einfachen Vergleich Vergleich der Modelle anhand von Genauigkeit und Effizienz möglich macht. Die beiden Werte, die in diesem Zusammenhang betrachtet werden, sind die mittlere durchschnittliche Präzision (Mean Average Precision (mAP)) und die Verlustfunktion (Loss).

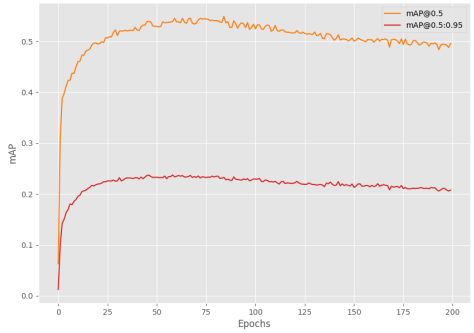
5.1.1 Mean Average Precision (mAP)

Die mittlere durchschnittliche Präzision (mAP) ist ein wichtiger Wert zur Bewertung der Leistungsfähigkeit von Objekt-Erkennungsmodellen. Sie kombiniert die Genauigkeit in Form von Intersection of Union (IoU) mit Precision und Recall der Erkennung in einem einzigen Wert und ermöglicht so einen Überblick über die Performance des Modells.

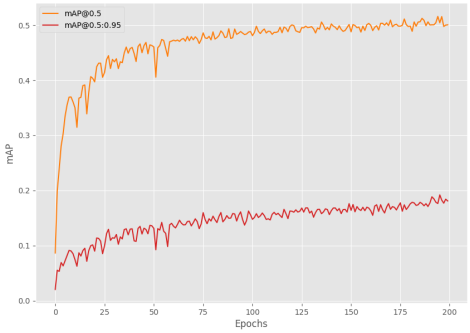
Betrachtet wird hier die mAP mit einer Intersection of Union (IoU) ab 0.5, da die Bounding Boxes in den Ground Truth Bildern meist mehr als nur die Korkpore beinhalten und somit auch größere Verschiebungen kein großes Problem darstellen.

Auffällig ist die Grafik von SSD. Im Vergleich zu allen anderen Modellen ist mAP deutlich schlechter. Bei den anderen Modellen beginnt mAP bei ca. 0,4. Das SSD-Modell hat jedoch seine beste Leistung nach ca. 30 Epochen bei 0,006. Dies ist auch in den folgenden Tests erkennbar. Das SSD-Modell erkennt in keinem der Tests Korkporen auf den Äpfeln. Der Vollständigkeit halber werden die Ergebnisse dennoch dargestellt. Um auszuschließen, dass die Ergebnisse aufgrund schlecht beschrifteter Bilder so niedrig sind, wurde das Modell mit einem anderen Datensatz getestet. Es zeigte sich, dass SSD mehr Bilder benötigt, um bessere Ergebnisse zu erzielen. Nach Tests mit einem anderen Datensatz mit ca. 900 Bildern lieferte das Modell ein deutlich besseres Ergebnis. Der gleiche Datensatz mit nur 100 Bildern erreichte wiederum einen ähnlich niedrigen Wert wie der Datensatz mit den Äpfeln.

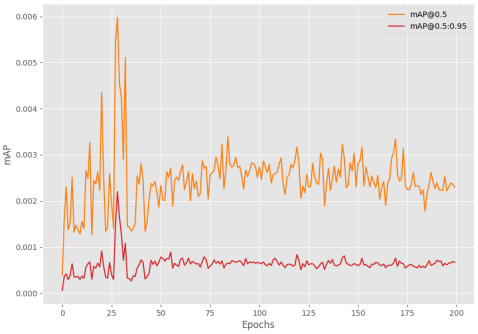
RetinaNet und Faster R-CNN haben beide eine mAP von ca. 0.5, wobei Faster R-CNN seine beste Leistung nach 50 Epochen erreicht und dann abfällt, während RetinaNet stetig ansteigt und seine beste Leistung nach den letzten Epochen erreicht.



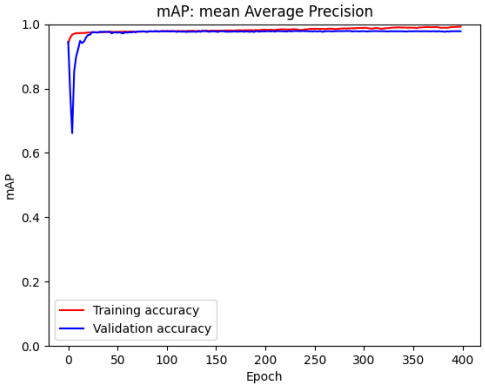
(a) Faster R-CNN



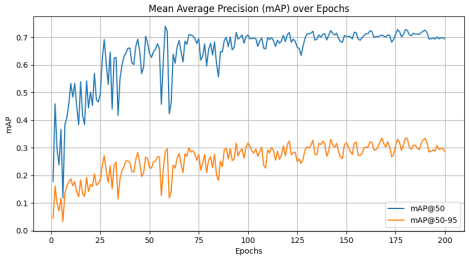
(b) RetinaNet



(c) SSD



(d) U-Net



(e) YOLO

Abbildung 5.1: Mean Average Precision (mAP)

YOLO zeigt in den ersten 75 Epochen starke Schwankungen und erreicht nach ca. 60 Epochen mit ca. 0.75 ebenfalls die beste mAP. Danach stagniert der Kurvenverlauf und zeigt keinen Anstieg mehr. Die höchste mAP erreicht U-Net mit fast 1. Auch in den folgenden Tests zeigt das Modell eine hohe Erkennungsrate. Die Tatsache, dass U-Net als einziges Modell einen so hohen Wert erreicht, zeigt, dass Segmentierungsmodelle möglicherweise besser geeignet sind als Objekterkennungsmodelle.

U-Net wurde speziell für die Erkennung kleiner Details in biomedizinischen Bildern entwickelt (Ronneberger, Fischer und Brox [2015]). Da auch die Korkporen kleine feine Details auf Äpfeln sind, ist es für das Segmentierungsmodell einfacher, mit einem so kleinen Datensatz sehr gute Ergebnisse zu erzielen. Die anderen Modelle sind als Objekterkennungsmodelle darauf spezialisiert, größere Objekte zu erkennen und zu klassifizieren. Daher benötigen sie mehr Bilder, um speziell für diese Anwendung trainiert zu werden. Nichtsdestotrotz haben auch diese Modelle gute Ergebnisse erzielt, aber in der mAP zeigen sich bereits die Vorteile, die U-Net mit sich bringt.

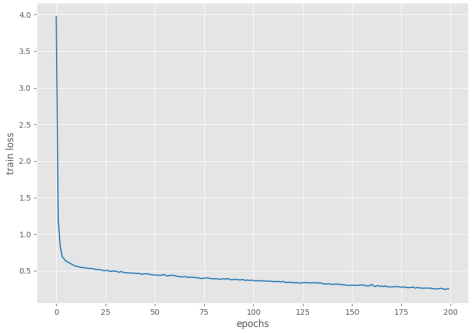
5.1.2 Loss

Der Verlust (Loss) ist ein wesentliches Element im Training von Objekterkennungsmodellen. Sie misst den Unterschied zwischen den vorhergesagten Werten und den tatsächlichen Werten und dient als Grundlage für die Optimierung des Modells.

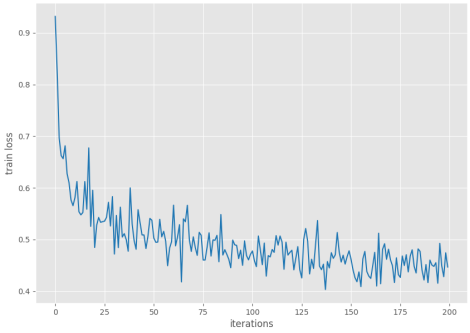
Der Verlust zeigt deutlich, dass das SSD-Modell nicht gut trainiert wurde. Mit einem Verlustwert von 4.5 weist es den mit Abstand höchsten und somit schlechtesten Wert auf. Dies deutet darauf hin, dass das Modell Schwierigkeiten hatte, die Trainingsdaten effektiv zu verarbeiten und zu generalisieren. Im Vergleich dazu erreicht YOLO einen Verlustwert von 1.5, der zwar deutlich besser ist als der von SSD, aber dennoch den zweit schlechtesten Wert unter den getesteten Modellen darstellt.

Die Modelle RetinaNet, Faster R-CNN, und U-Net liegen hingegen mit Verlustwerten von 0.45, 0.25 und 0.05 relativ nahe beieinander. U-Net erreicht dabei den besten Wert und übertrifft die anderen Modelle. Dies lässt darauf schließen, dass U-Net die Trainingsdaten am effizientesten verarbeitet hat und somit eine deutlich bessere Performance zeigt.

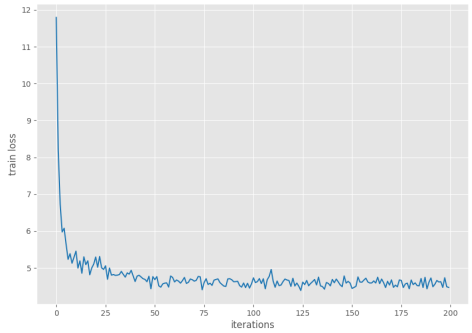
Die Erkenntnisse aus Unterabschnitt 5.1.1 lassen sich auf diese Ergebnisse übertragen und werden durch die hier präsentierten Verlustwerte weiter bestätigt. Es zeigt sich erneut,



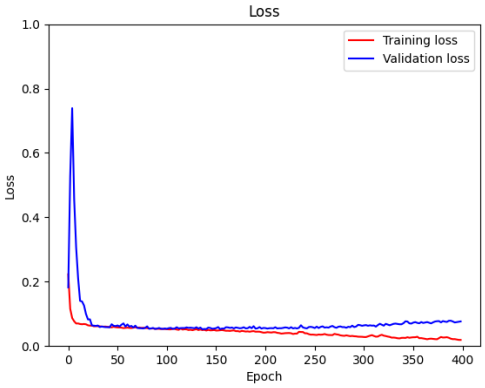
(a) Faster R-CNN



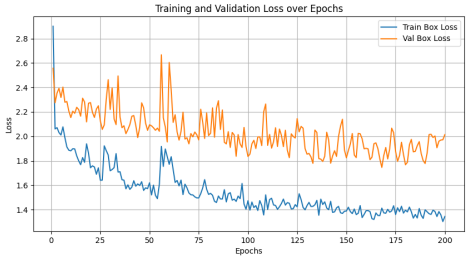
(b) RetinaNet



(c) SSD



(d) U-Net



(e) YOLO

Abbildung 5.2: Loss

dass U-Net insbesondere im Training eine überlegene Leistung erbringt. Interessanterweise weichen die Verlustwerte der anderen Modelle jedoch weniger stark von den Ergebnissen des U-Net ab als dies bei den in Unterabschnitt 5.1.1 dargestellten Metriken der Fall ist. Dies könnte darauf hindeuten, dass diese Modelle während des Trainings gewisse Aspekte der Daten dennoch gut gelernt haben, auch wenn sie insgesamt hinter den Resultaten von U-Net zurückbleiben.

5.2 Apfelsorten

In diesem Abschnitt werden die verschiedenen Apfelsorten beschrieben, die im Rahmen dieser Arbeit untersucht und klassifiziert wurden. Dabei wird zwischen den Apfelsorten, die zum Trainieren verwendet wurden, und den Apfelsorten, mit denen das Modell keine Berührungspunkte hatte, unterschieden. Diese Differenzierung ist wichtig, um zu beurteilen, wie robust das Modell gegen unbekannte Sorten ist. So wird festgestellt, inwiefern das Modell generalisierbar ist.

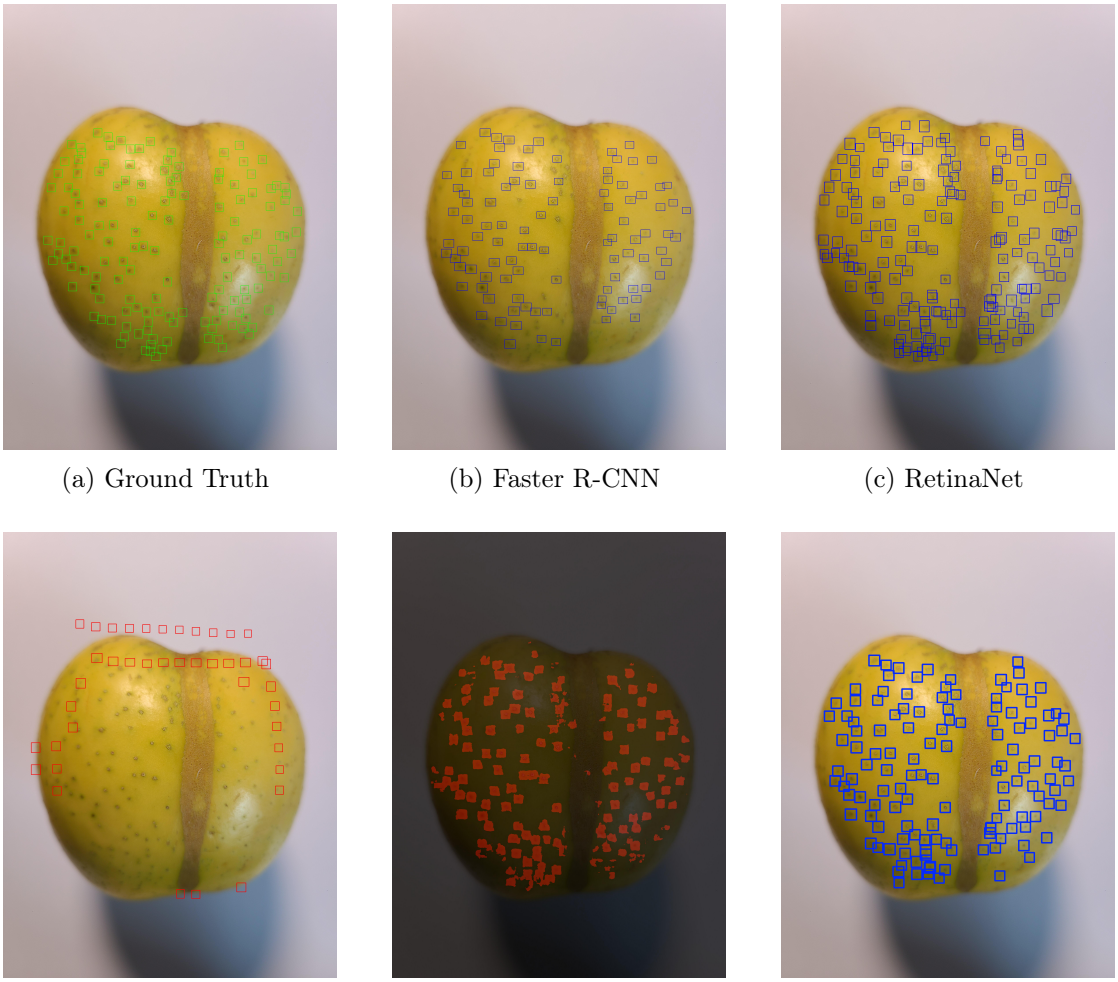
5.2.1 Trainierte Apfelsorten

Die trainierten Apfelsorten umfassen die Apfelsorten, die während des Trainingsprozesses des Modells verwendet wurden. Dazu gehören Golden Delicious und Wellant. Bei beiden Äpfeln sind die Korkporen in der Regel gut sichtbar, was die Erkennung erleichtert.

Golden Delicious

Die Farbe des Apfels ist einheitlich grün, daher sind auch die Korkporen einheitlich braun gefärbt.

Wie erwartet liefert SSD schlechte Ergebnisse. Dies wird sich in den nächsten Tests fortsetzen, daher konzentriert sich dieser Abschnitt auf die anderen vier Modelle. Mit einem Apfel, der dem aus dem Trainingsdatensatz sehr ähnlich ist, schneiden alle sehr gut ab. Faster R-CNN erkennt jedoch viele Korkporen an den Rändern rechts unten und oben nicht. An diesen Stellen werden die Korkporen kleiner und verschwimmen etwas. YOLO hat auch einige False Negatives unten rechts.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.3: Golden Delicious

RetinaNet hingegen hat einige False Positives, die sich an den Rändern häufen. Viele Bounding Boxes umschließen die gleiche Korkpore, was darauf hindeuten könnte, dass das Modell zu oft mit den gleichen Daten trainiert wurde, also ein Overfitting stattgefunden hat. Es kann auch sein, dass die Bounding Boxes der Ground Truth Bilder nicht genau genug gesetzt wurden. Wie in Abbildung 5.3(a) zu sehen ist, überlappen sich einige Bounding Boxes, was zu einem fehlerhaften Training geführt haben könnte. Dagegen spricht, dass die anderen Modelle nur sehr wenige False Positives aufweisen. Auch bei YOLO gibt es einige Bounding Boxes, die die gleichen Korkporen umfassen, aber deutlich weniger als bei RetinaNet.

U-Net erzielt auch in den bereits hervorgehobenen kritischen Bereichen gute Detektionen. Gerade unten werden auch die kleinen Korkporen erfolgreich detektiert. False Positives gibt es zwar auch, aber das sind meist die Detektionen, die nur ein oder zwei Pixel umfassen.

'Bekannter' Wellant

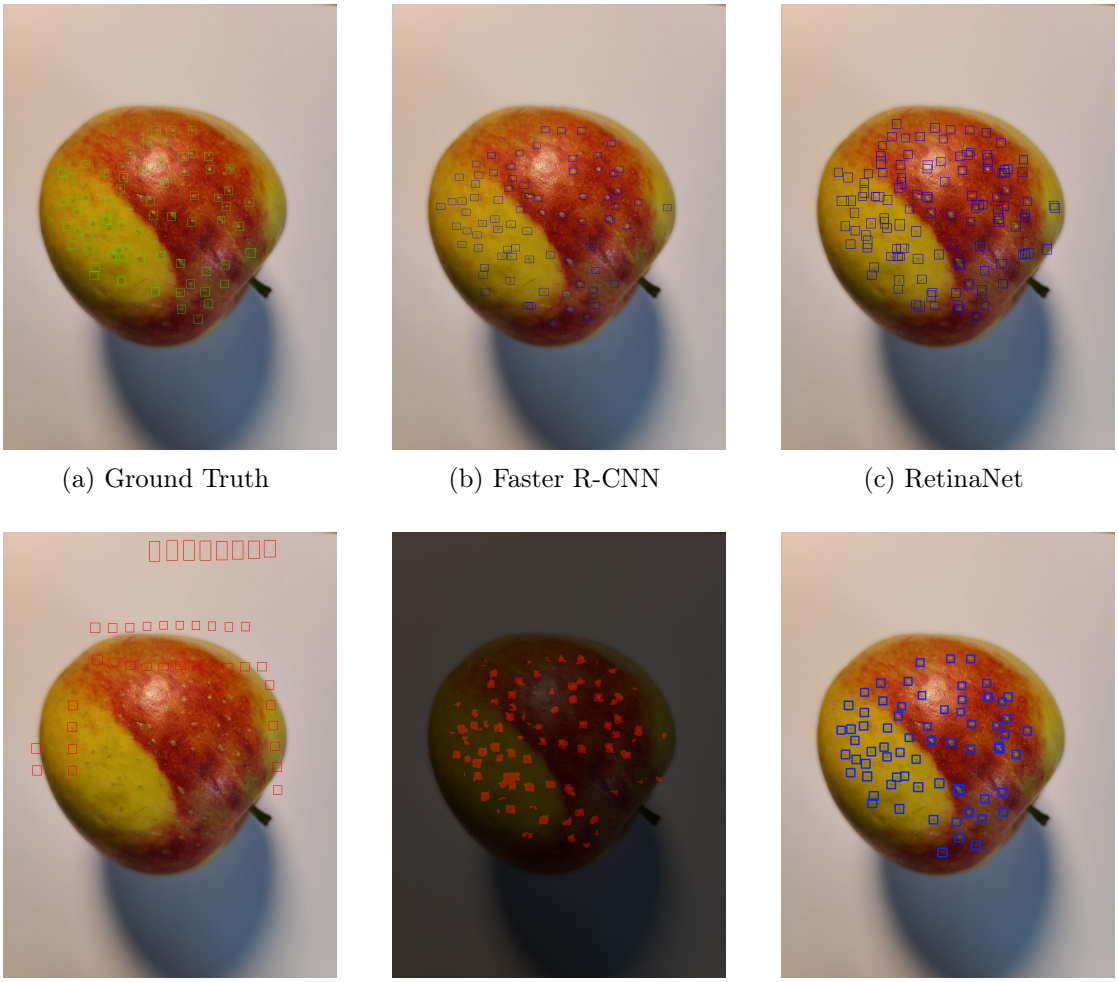
Die Farbe von Wellant variiert zwischen grünlich und rötlich. Dementsprechend sind auch die Korkporen unterschiedlich gefärbt, aber immer noch gut erkennbar. Zum einen wurden die Modelle mit einem Bild verglichen, das vor dem Training aus dem Trainingssatz entnommen wurde und daher den Trainingsbildern sehr ähnlich ist.

Auch bei diesem Bild erkennt RetinaNet fast alle Korkporen, hat aber wieder einige False Positives, die sich um die True Positives gruppieren. Die anderen drei Modelle haben alle eine gute Erkennungsrate. Faster R-CNN erkennt jedoch keine Korkporen auf der hellen Stelle, wo das Licht reflektiert wird. Die anderen Modelle können dies. Die Lichtverhältnisse sind hier im Vergleich zum Rest des Apfels deutlich anders, die Farbe der Korkporen und ihre Umgebung unterscheiden sich deutlich, so dass das gelernte Verhalten des Modells in diesem Bereich schwieriger anzuwenden ist. Außerdem gibt es hier nur wenige Trainingsfälle, da der Bereich sehr klein ist und häufig keine Korkporen vorhanden sind.

Weiterhin ist auch hier auffällig, dass die Modelle wieder Probleme mit Korkporen am Rand des Apfels haben. In Abbildung 5.4(e) sieht man unten rechts, dass um den Stiel des Apfels herum kaum Detektionen gemacht wurden. Dies deutet darauf hin, dass dieser Fall im Trainingsdatensatz nicht ausreichend abgedeckt wurde. Da die Korkporen in diesen Bereichen auch für das menschliche Auge schwer zu erkennen sind, kann es vorkommen, dass im Trainingsdatensatz einige übersehen wurden. Dies führt dann zu Inkonsistenzen in der späteren Detektion.

'Fremder' Wellant

Andererseits wurde ein Bild von Wellant in den Testdatensatz aufgenommen, das neu aufgenommen wurde und daher eine eigene Beleuchtung und einen eigenen Hintergrund



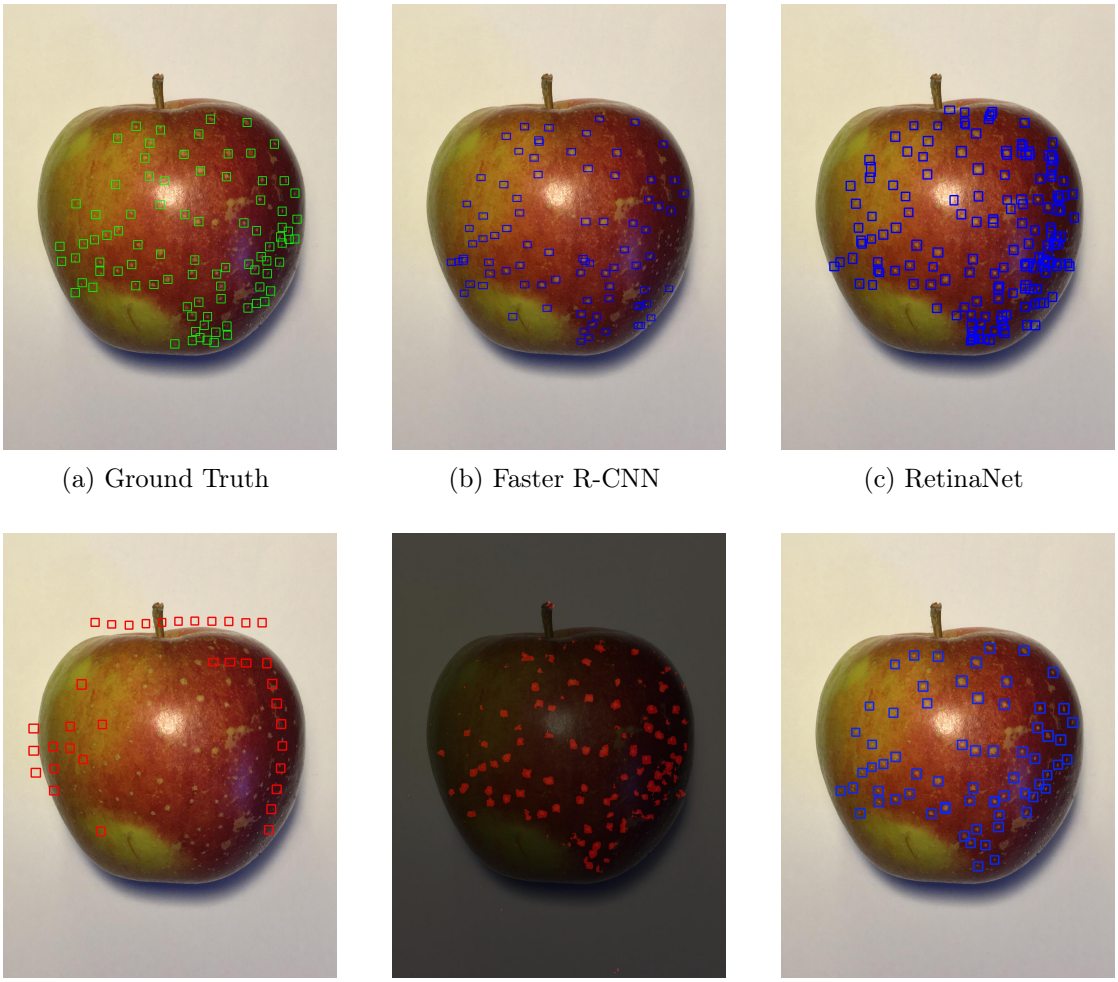
(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.4: 'Bekannter' Wellant

hat. So können Unterschiede zwischen bekannten und unbekannten Bildern herauskristallisiert werden, da die gleiche Sorte verwendet wird. Dieser Apfel dient später auch als Grundlage für die anderen Testkategorien.

Auch hier treten die gleichen Probleme auf. RetinaNet hat wieder viele False Positives, vor allem am rechten unteren Rand von Wellant, aber auch an anderen Stellen. Dagegen erkennen YOLO und U-Net nur wenige Korkporen am rechten unteren Rand, haben aber ansonsten eine gute Erkennungsrate. Faster R-CNN hingegen erkennt auch an dieser kritischen Stelle noch die meisten Korkporen. Je weiter man sich jedoch dem Rand nähert, desto schwieriger wird es auch für dieses Modell. Auffällig ist, dass YOLO an diesem



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.5: 'Fremder' Wellant

rechten Rand gar keine Korkporen erkennt, Faster R-CNN zwar am rechten unteren Rand, aber nicht am rechten Rand und U-Net am rechten Rand, aber dann nicht unten rechts in der Ecke.

Ein weiteres Problem zeigt sich auch hier wieder. Der reflektierende Bereich, hier ziemlich in der Mitte des Apfels, erschwert die Erkennung in diesem Bereich. Faster R-CNN hat wieder Schwierigkeiten, dort eine Korkpore zu erkennen, aber auch YOLO und RetinaNet, das zwar eine erkennt, die aber, wie in Abbildung 5.5(a) zu sehen, gar keine ist. U-Net erkennt hier als einziges Modell genau die Korkporen, die auch im Ground Truth zu

finden sind. Tatsächlich erweist sich U-Net in diesem Bild als das genaueste, auch wenn unten rechts am Rand viele Korkporen übersehen wurden.

5.2.2 Unbekannte Apfelsorten

Die unbekannten Apfelsorten beziehen sich auf jene Sorten, die nicht im Trainingsdatensatz enthalten waren und ausschließlich zum Testen herangezogen wurden. Die Fähigkeit des Modells, diese unbekannten Apfelsorten korrekt zu erkennen und zu klassifizieren, ist ein wesentlicher Indikator für seine Leistungsfähigkeit und Generalisierung. Zu den gewählten Sorten zählen Crimson Snow, Elstar, Granny Smith und Red Jonaprince. Die Auswahl dieser Sorten wurde so getroffen, dass ein möglichst breites Spektrum an Merkmalen getestet werden kann. So gibt es grüne Äpfel, Äpfel mit schwer erkennbaren Korkporen oder Äpfel, deren Korkporen unterschiedlich gefärbt sind. So wird die Möglichkeit geschaffen, fundierte Erkenntnisse aus den Tests zu ziehen.

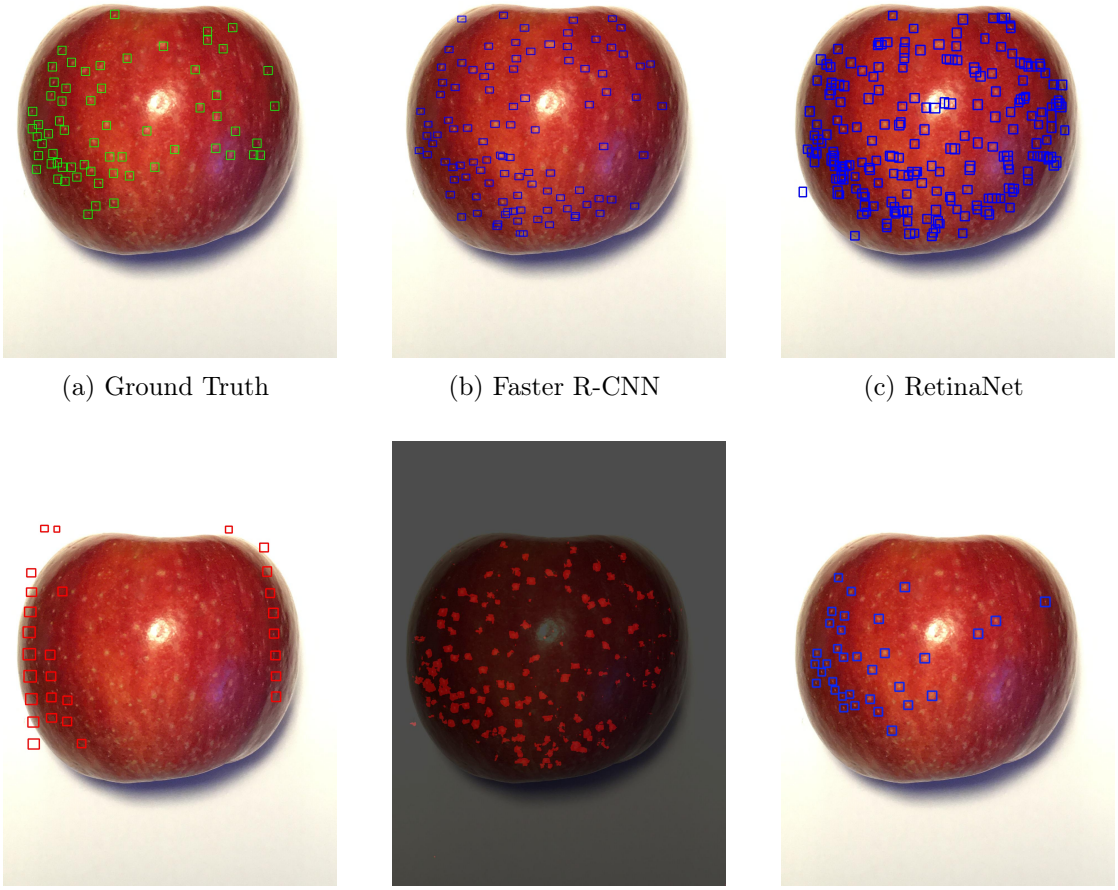
Crimson Snow

Merkmale dieser Sorte sind die rote Haut und das weiße Fruchtfleisch. Auch die Korkporen sind sehr hell und eher klein.

Der Crimson Snow zeigt erneut das Problem der menschlichen Erkennung, das in Unterunterabschnitt 5.2.1 erwähnt wurde. Es ist sehr schwierig zu entscheiden, was als Korkpore und was nur als Farbveränderung auf der Schale des Apfels anzusehen ist. Zur Festlegung einheitlicher Kennzeichnungen wurde daher die Intensität des Flecks herangezogen. Die in Abbildung 5.6(a) markierten Korkporen sind deutlich heller als die Flecken im unteren Bereich. Daher wurden nur diese als Korkporen identifiziert und markiert.

Diese Unterscheidung spiegelt sich auch in den Modellergebnissen wider. YOLO liegt mit seinen Detektionen am nächsten an Ground Truth, erkennt aber auch in diesem Vergleich einige Korkporen nicht. Faster R-CNN und U-Net hingegen erkennen diese helleren Punkte im unteren Bereich des Apfels als Korkporen. U-Net erweist sich in dieser Wahrnehmung als konsistenter, da Faster R-CNN einige Flecken übersieht.

Die oben genannten Probleme bleiben bestehen. RetinaNet erkennt alle Flecken als Korkporen, hat aber wieder viele False Positives. Auch Faster R-CNN und U-Net übersehen wieder Korkporen an den Rändern des Apfels.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

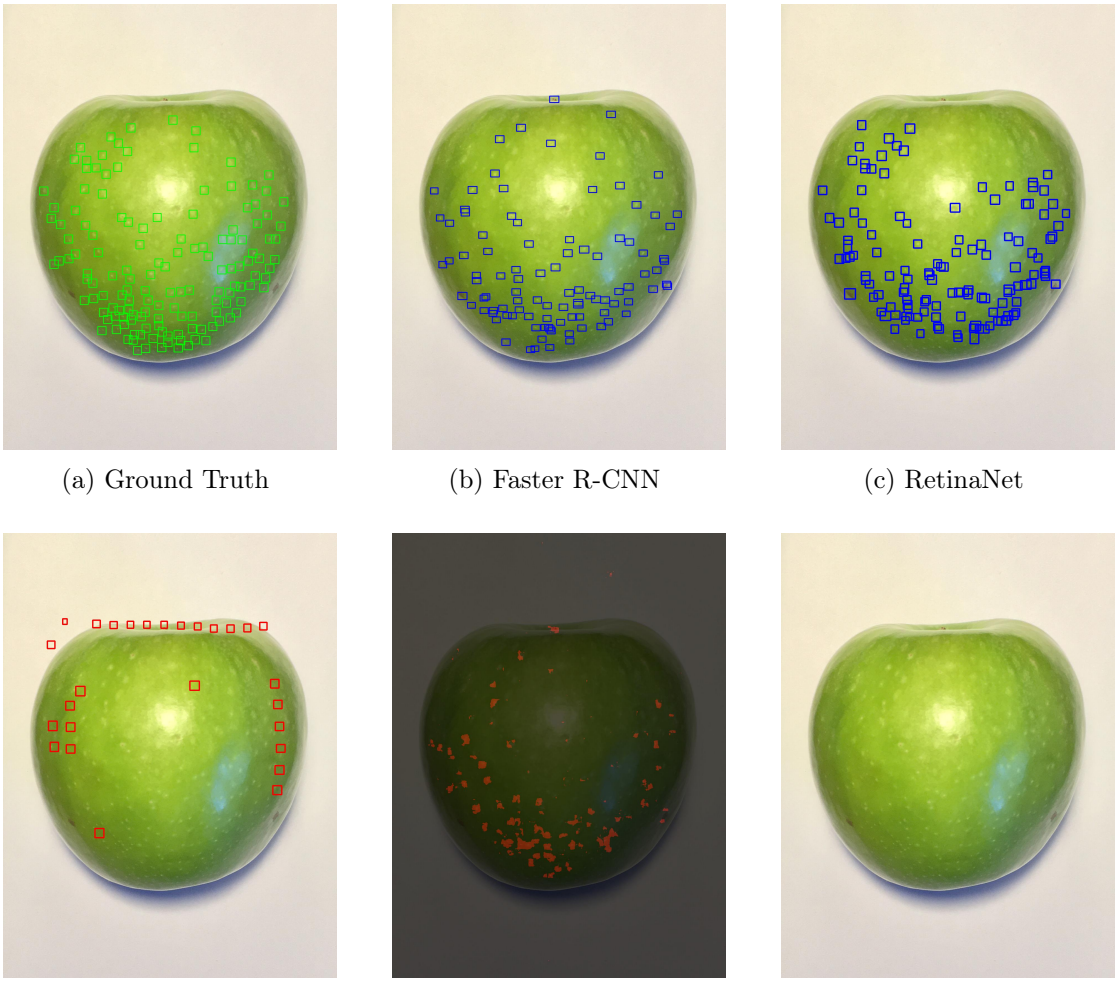
(d) SSD (e) U-Net (f) YOLO

Abbildung 5.6: Crimson Snow

Betrachtet man die Ergebnisse in Relation dazu, dass die Modelle einen ihnen unbekannten Apfel verarbeitet haben, kann man schlussfolgern, dass das Konzept der Korkpore gut gelernt wurde. Die auffälligsten Merkmale, der Farbunterschied und die Größe, wurden von den Modellen gut gelernt. Bei Crimson Snow ist jedoch auch hilfreich, dass die Korkporen wie bei Wellant hell auf rotem Hintergrund waren.

Granny Smith

Im Handel ist diese Apfelsorte in einem knalligen Grün zu finden. Die Korkporen sind sehr hell und gut erkennbar.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net granny-smith (f) YOLO

Abbildung 5.7: Granny Smith

Der Granny Smith weist mit seiner Farbe und Struktur keine Ähnlichkeit zu dem Wellant oder Golden Delicious auf. Im Gegensatz zu den trainierten Sorten, sind die Korkporen heller als die Haut und nicht wie bei Wellant auf roter sondern auf grüner Oberfläche verteilt.

Wie in Abbildung 5.7 zu sehen, haben alle Modelle starke Schwierigkeiten mit dieser Sorte. Hervorzuheben ist da YOLO. Durch keine einzige Detektion wird klar, dass die gelernten Strukturen und Farbunterschiede sich hier nicht anwenden lassen. Die anderen drei Modelle erkennen immerhin im unteren Bereich des Apfels noch einige Korkporen, je heller die Oberfläche aber wird, desto weniger Detektionen sind zu sehen. Der Helligkeitsunterschied scheint zu gering zu sein.

Die schwachen Ergebnisse lassen sich also darauf zurückführen, dass zum einem der Apfel den trainierten Sorten gegenüber zu unähnlich war und zum anderen, die Korkporen dem Rest des Apfels zu ähnlich, also nicht genug hervor gestochen sind.

Elstar

Der Elstar hat meist eine gelbliche Haut mit roten Stellen, die auf der von der Sonne bestrahlten Seite entstehen. Korkporen sind eher schwer zu erkennen.

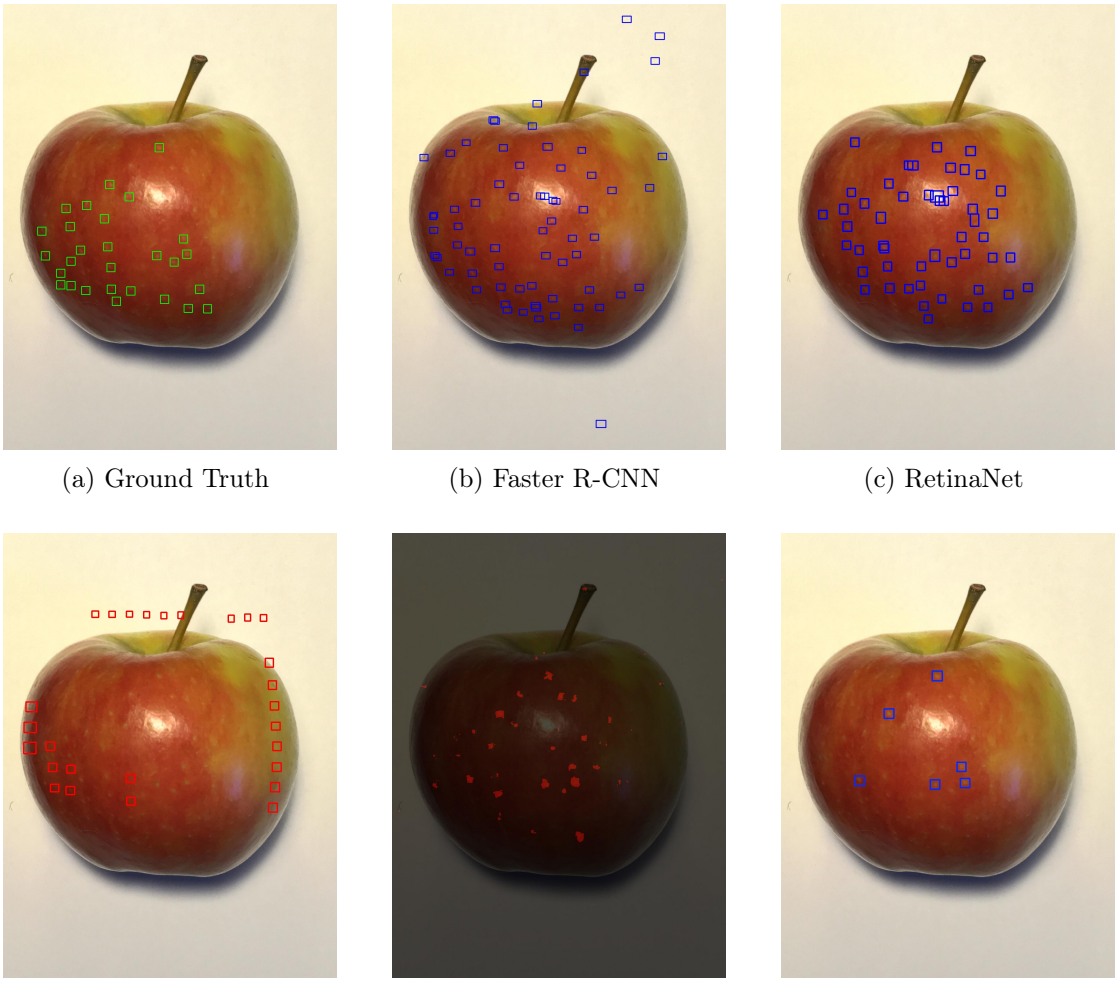
Mit dieser Sorte wurde ein Apfel getestet, dessen Korkporen nur sehr schlecht zu erkennen sind. Faster R-CNN und RetinaNet haben die höchste Erkennungsrate, aber beide haben auch viele False Positives. U-Net hingegen hat viele False Negatives und die False Positives sind wiederum sehr gering. YOLO schneidet wie in Unterunterabschnitt 5.2.2 nicht gut ab und erkennt kaum Korkporen.

Die Ergebnisse von Unterunterabschnitt 5.2.2 finden sich zum Teil auch bei Elstar. Hinzu kommt jedoch, dass die Korkporen auch für das menschliche Auge nur schwer zu erkennen sind, die gelernten Strukturen der Modelle also auch hier nicht wirklich anwendbar sind.

Red Jonaprince

Der Red Jonaprince zeichnet sich wie der Crimson Snow durch seine sehr rote Haut aus. Mit kleinen und hellen Korkporen ähnelt er auch hier dem Crimson Snow, die allerdings deutlich besser zu erkennen sind.

Durch die deutlich besser erkennbaren Korkporen sind die Detektionen deutlich besser als bei Crimson Snow. Insbesondere U-Net und YOLO erkennen viele Korkporen. Auch das bekannte Problem der Apfelränder wird bei Jonaprince gut gelöst. Durch ihre Helligkeit und den intensiven roten Hintergrund heben sich die Korkporen auch am Rand deutlich von der Schale ab und sind somit auch dort gut zu erkennen.



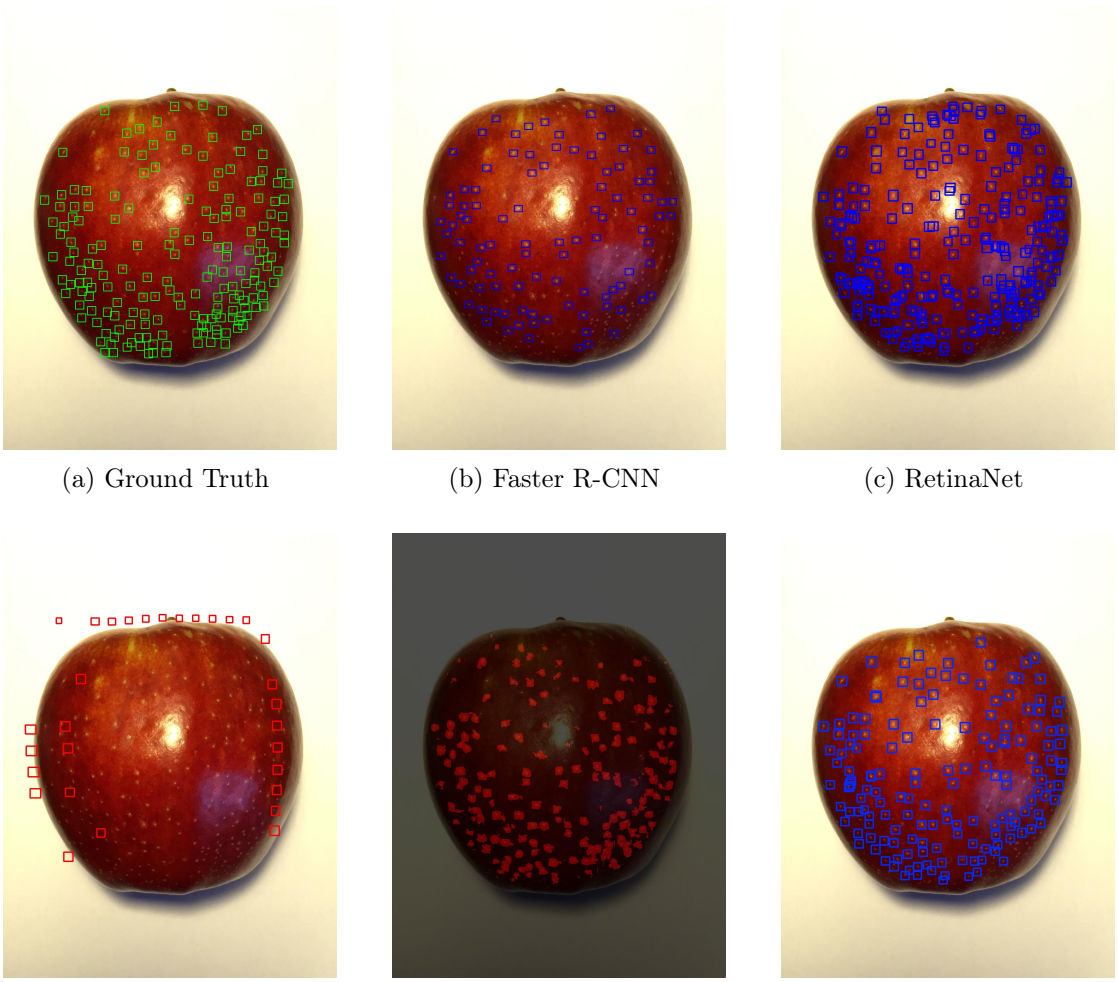
(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.8: Elstar

Auch Faster R-CNN und RetinaNet erkennen einige Korkporen. RetinaNet findet zwar auch viele am Rand, aber wieder verbunden mit einer hohen Anzahl von False Positives, so dass es ein wenig wie Raten aussieht. Faster R-CNN hingegen hat nach wie vor große Schwierigkeiten am Rand des Apfels. Im Vergleich zu Ground Truth findet das Modell in diesem Bereich sehr wenige Korkporen.

Grundsätzlich kann man aber sagen, dass die Modelle auch mit den unbekannten Sorten gut zurecht kommen. In Fällen wie Granny Smith, wo die Struktur völlig unbekannt war, hatten sie alle Schwierigkeiten. Allerdings scheint die Struktur einer Korkpore mit den 100 Trainingsbildern ausreichend vermittelt worden zu sein, um auch Korkporen auf un-



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.9: Red Jonaprince

bekannten Äpfeln zu erkennen. Besonders hervorzuheben ist U-Net, das im Gegensatz zu den anderen Modellen nur wenige Probleme hatte und insgesamt Korkporen zuverlässig erkennen konnte.

5.3 Bildqualität

In diesem Abschnitt werden verschiedene Aspekte der Bildverarbeitung behandelt. Rauschen und Downsampling, die häufigsten Probleme in der Bildverarbeitung, werden genauer betrachtet. Die Modelle werden mit verschiedenen Varianten und Stärken vergli-

chen. Dies gibt einen guten Überblick über ihre Robustheit gegenüber Störungen im Bild.

5.3.1 Rauschen

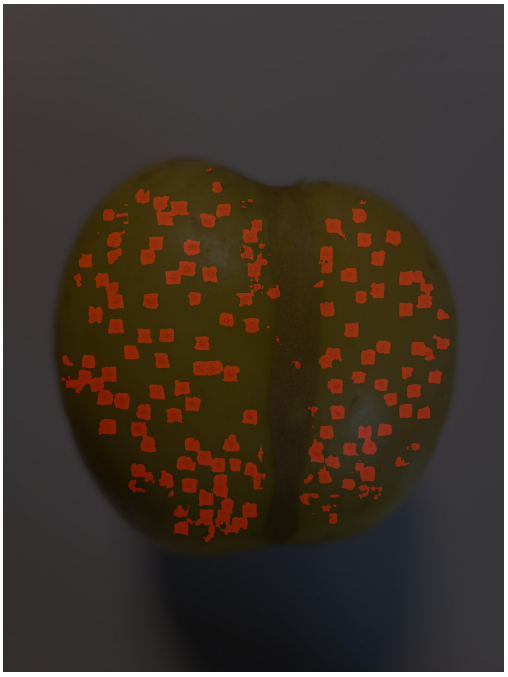
Für das Rauschen werden die beiden bereits erläuterten Varianten Gaußsches Rauschen und Salz-und-Pfeffer-Rauschen zum Testen der Modelle verwendet. Dabei stehen verschiedene Intensitätsstufen zur Verfügung. So kann festgestellt werden, ab welcher Rauschintensität die Modelle jeweils an ihre Grenzen stoßen.

Gaussches Rauschen

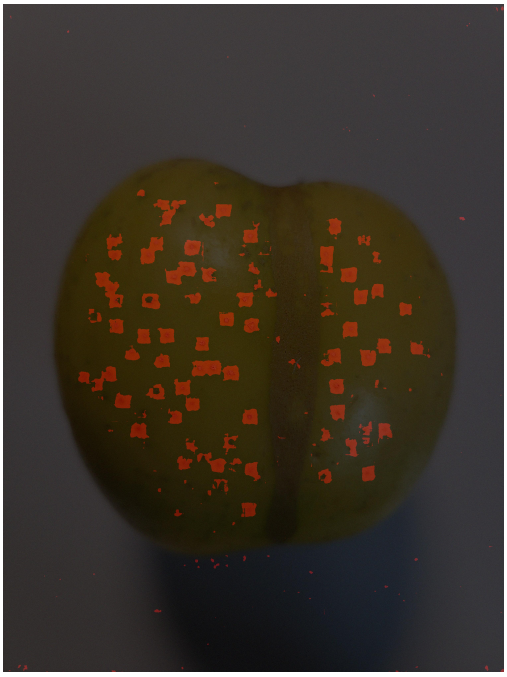
Die Ergebnisse sind für U-Net und YOLO dargestellt. Die Entwicklung der Detektionen von U-Net spiegelt die Entwicklung der anderen Modelle gut wider. YOLO hingegen kommt am besten mit dem Gaußschen Rauschen zurecht.

Bei Abbildung 5.10 ist zu erkennen, dass die Anzahl der Detektionen mit steigender Standardabweichung abnimmt. Dies kann darauf zurückgeführt werden, dass, wie in Unterunterabschnitt 4.5.1 erwähnt, die Intensität der Pixel verändert wird. Die gelernten Farbverhältnisse sind mit steigender Intensität schwieriger anzuwenden, was die sinkende Erkennungsrate erklärt. Bei U-Net ist zu erkennen, dass bei einer Standardabweichung von 0.1 zwar weniger Korkporen erkannt werden, das Modell aber immer noch fast ausschließlich Korkporen erkennt. Ab einer Standardabweichung von 0.25 ändert sich dies. In Abbildung 5.10(c) und (d) werden kaum noch Korkporen erkannt, dafür aber mehr außerhalb des Apfels. Insbesondere Abbildung 5.10(d) wirkt wie zufällig ausgewählte Pixel. Die Farbänderung der Pixel ist zu stark, als dass das Modell mit den gelernten Parametern noch erfolgreich Korkporen identifizieren könnte.

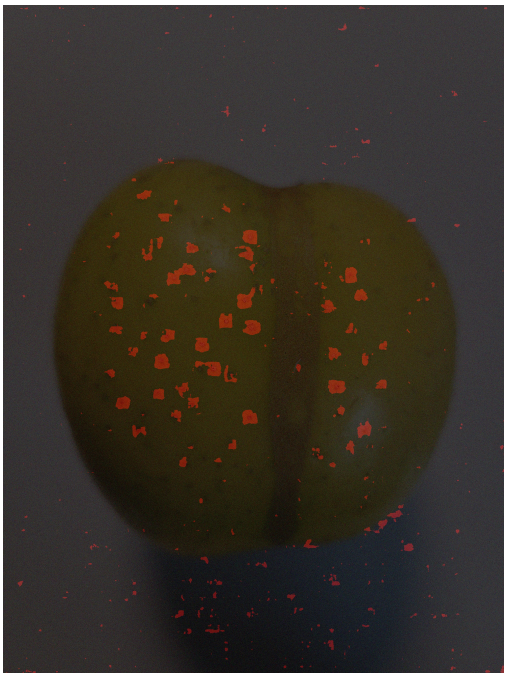
Auch YOLO bekommt mit zunehmender Intensität größere Schwierigkeiten. Auffällig ist aber der unterschiedliche Umgang. Während U-Net zunehmend zufällig wirkende Pixel oder Pixelgruppen als Korkporen identifiziert, werden die Detektionen von YOLO einfach nur weniger. Auch YOLO erkennt schon bei einer Standardabweichung von 0.1 weniger Korkporen, allerdings sind ausnahmslos alle Detektionen True Positives, wo hingegen U-Net auch einige False Positives hatte. Deutlicher wird das in Abbildung 5.11(c) und (d). YOLO hat keine False Positives. Das Modell scheint sehr gut gelernt zu haben, was als Korkpore eingestuft wird und was nicht. Auch in Unterunterabschnitt 5.2.2 und



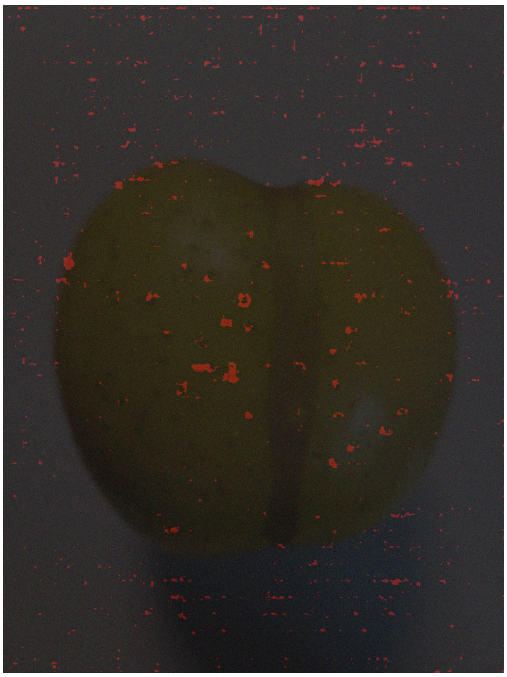
(a) Bild ohne additives Rauschen



(b) Additives Rauschen von 0.1

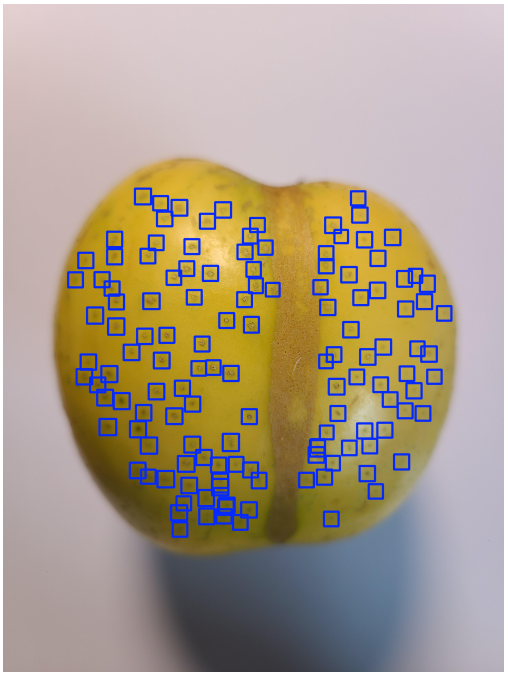


(c) Additives Rauschen von 0.25

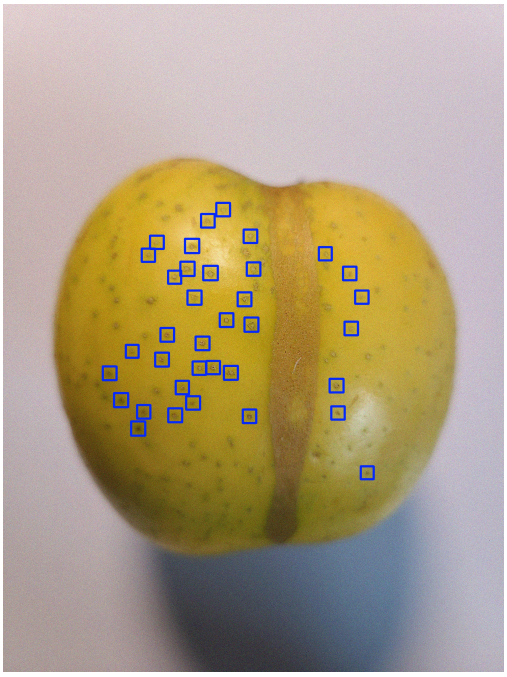


(d) Additives Rauschen von 0.5

Abbildung 5.10: U-Net mit verschieden starken Standardabweichungen



(a) Bild ohne additives Rauschen



(b) Additives Rauschen von 0.1

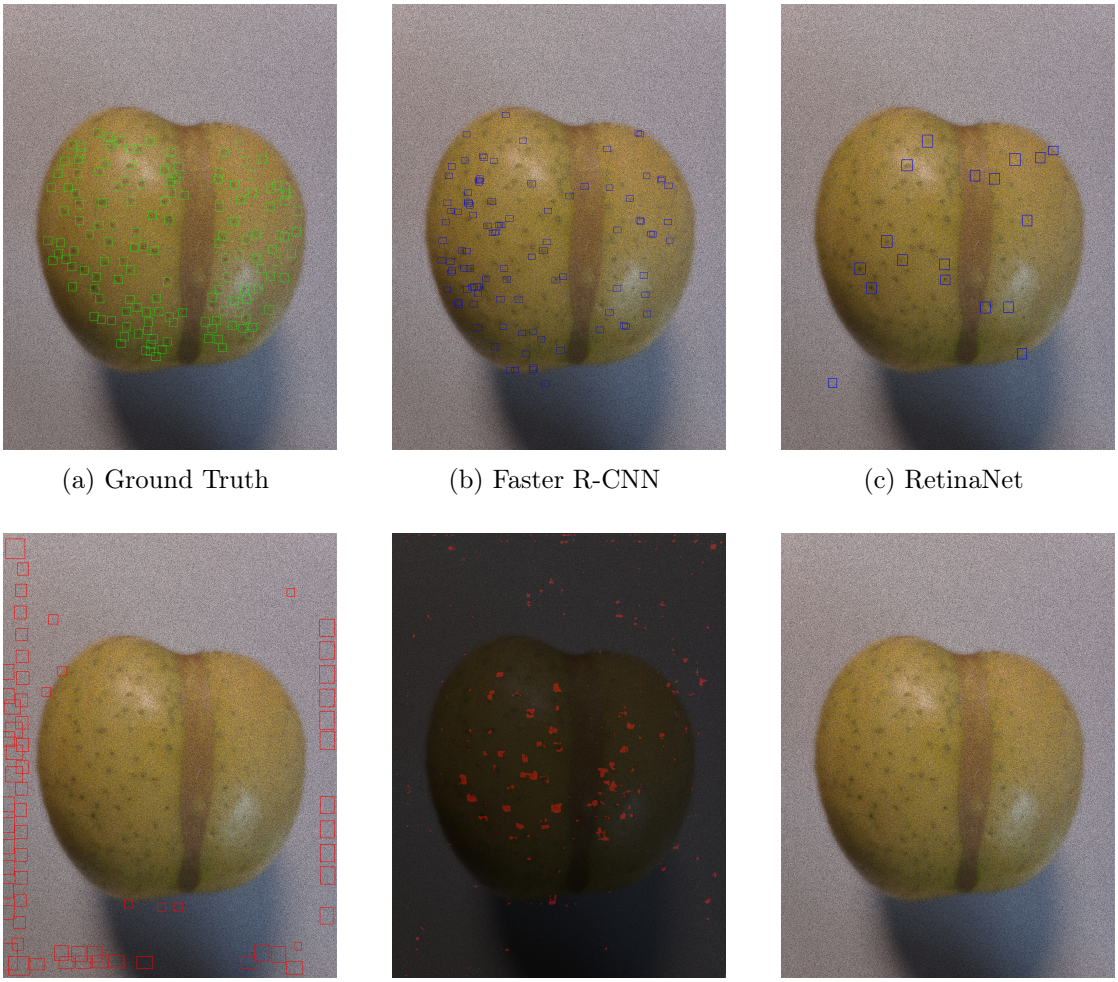
(c) Additives Rauschen von 0.25

(d) Additives Rauschen von 0.5

Abbildung 5.11: You Only Look Once (YOLO) mit verschieden starken Standardabweichungen

Unterunterabschnitt 5.2.2 findet sich diese Erkenntnis wieder. Wo andere Modelle zu False Positives neigen, scheint YOLO ein gutes Verständnis dafür entwickelt zu haben, was eine Korkpore ausmacht. Das macht es robuster gegenüber Fehlerkennungen.

Salz-und-Pfeffer Rauschen



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.12: Salz-und-Pfeffer Rauschen mit einer Rauschdichte von 0.1

Die Ergebnisse der Modelle mit Salz-und-Pfeffer Rauschen ähneln denen aus Unterunterabschnitt 5.3.1. Je dichter das Rauschen wird, desto größere Schwierigkeiten haben die Modelle Korkporen zu erkennen. Da der Verlauf dem des Gausschen Rauschens ähnelt, wird dieser nicht erneut bildlich dargestellt. Stattdessen soll Abbildung 5.12 die grund-

sätzlichen Probleme und Merkmale der Modelle verdeutlichen. YOLO hat erneut keine False Positives, die anderen Modelle hingegen schon. Das zieht sich durch alle Ergebnisse mit verschiedenen Rauschdichten. Auch die Detektionen, die RetinaNet und Faster R-CNN getroffen haben sind nur selten korrekt. Über alle Modelle lässt sich also sagen, dass sie bei Bildern mit Salz-und-Pfeffer Rauschen nur sehr wenig True Positives haben.

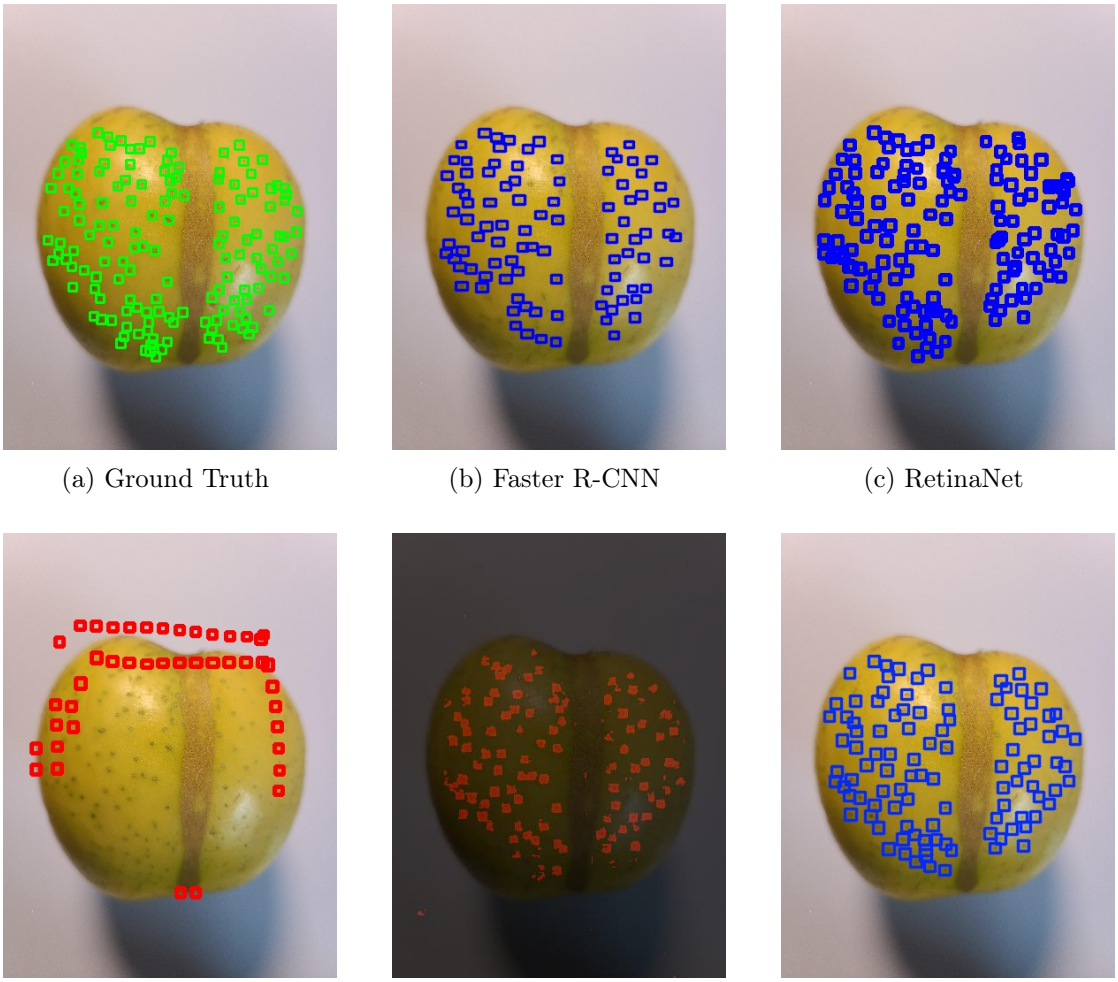
Bei einer Rauschdichte von 0.1 wird ungefähr jeder zehnte Pixel entweder schwarz oder weiß, wie in Unterunterabschnitt 4.5.1 beschrieben wurde. Die Erkenntnis aus Unterunterabschnitt 5.3.1, dass die Modelle die Detektion stark von den Farben und Farbdifferenzen der relevanten Bereiche abhängig machen, wird mit diesen Ergebnisse gestützt. Da sich bei dieser Art Rauschen nicht die Intensität ändert, sondern der Pixel ganz schwarz oder weiß wird, ist der Unterschied zu den bekannten Korkporen beträchtlicher. Das führt dazu, dass das gelernte Farbbild des Apfels und seiner Korkporen mit dieser Art von Rauschen gar nicht anwendbar ist.

5.3.2 Downsampling

Beim Downsampling werden vier verschiedene Größen mit den Skalierungsfaktoren $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ und $\frac{1}{16}$ betrachtet. Diese Methode simuliert Szenarien, in denen die Bildqualität der aufgenommenen Bilder deutlich schlechter ist als die der trainierten Bilder. Da die Ergebnisse über die verschiedenen Größen hinweg ähnlich sind, werden sie anhand der Bilder mit einer Skalierung von $\frac{1}{8}$ dargestellt.

In Abbildung 5.13 lässt sich erkennen, dass trotz deutlich kleinerer Bilder die Erkennungsrate weiter hoch bleibt und vor allem ähnlich der Erkennungsrate der Modelle bei normaler Größe ist. Die üblichen Probleme treten wieder auf. RetinaNet hat einige False Positive und die übrigen Modelle übersehen an den Rändern des Apfels erneut Korkporen. Auch U-Net hat in diesen Bereichen Schwierigkeiten.

Auch die Erkenntnis aus diesem Test steht mit den restlichen Erkenntnissen aus diesem Unterkapitel im Konsens. Da bei Runterskalierung weder Farbe noch Form verändert werden, leidet die Erkennungsrate auch nicht unter solchen Bildveränderungen.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

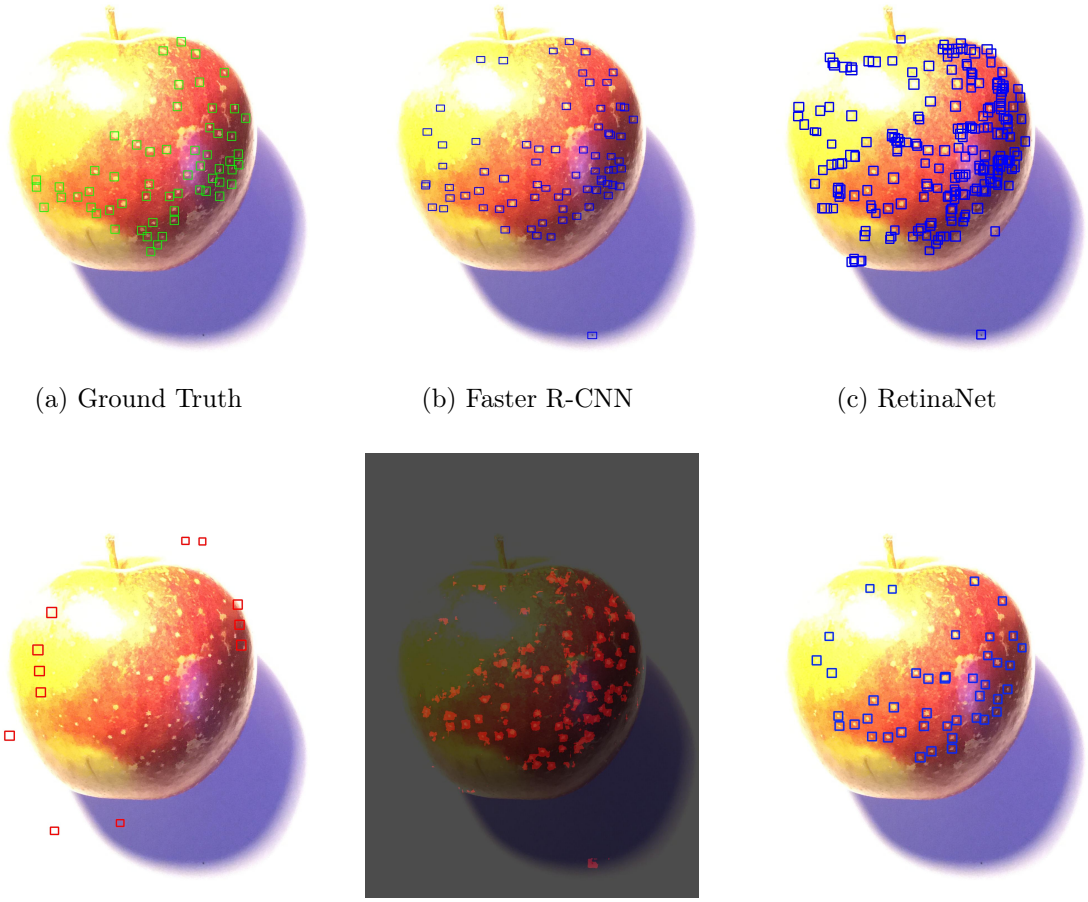
(d) SSD (e) U-Net (f) YOLO

Abbildung 5.13: Herunterskalierung mit einem Faktor von $\frac{1}{8}$

5.4 Beleuchtung

Die Beleuchtung hat einen erheblichen Einfluss auf die Bildqualität und die Erkennungsleistung von Modellen. Unterschiedliche Lichtverhältnisse, wie schwaches Licht oder blendendes Licht, können zu variierenden Helligkeiten und Kontrasten führen. In dieser Testkategorie wird untersucht, wie gut das Modell auf veränderte Beleuchtungssituationen reagiert, die von den Bedingungen der Trainingsbilder abweichen.

Die Ergebnisse mit den hellen und dunklen Beispielbildern waren ähnlich und werden daher anhand des stark beleuchteten Apfels veranschaulicht.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.14: Stark beleuchteter Wellant

Alle Modelle erkennen die meisten Korkporen auf der roten, dunkleren Seite des Apfels. RetinaNet erkennt wieder viele False Positives. U-Net, YOLO und Faster R-CNN zeigen dagegen überwiegend True Positives, wobei U-Net viele kleine Spots markiert, die keine Korkporen sind. Dabei handelt es sich aber meist um kleine Verfärbungen.

Auffällig sind die Detektionen im hellen Bereich. Das menschliche Auge kann dort keine Korkporen erkennen, und doch haben alle Modelle an den gleichen Stellen Korkporen erkannt. In Abbildung 5.15, das aus der gleichen Position aufgenommen wurde, wird deutlich, dass sich an den fünf Stellen, die von YOLO und Faster R-CNN im hellen Bereich erkannt wurden, tatsächlich Korkporen befinden. Daraus kann geschlossen wer-

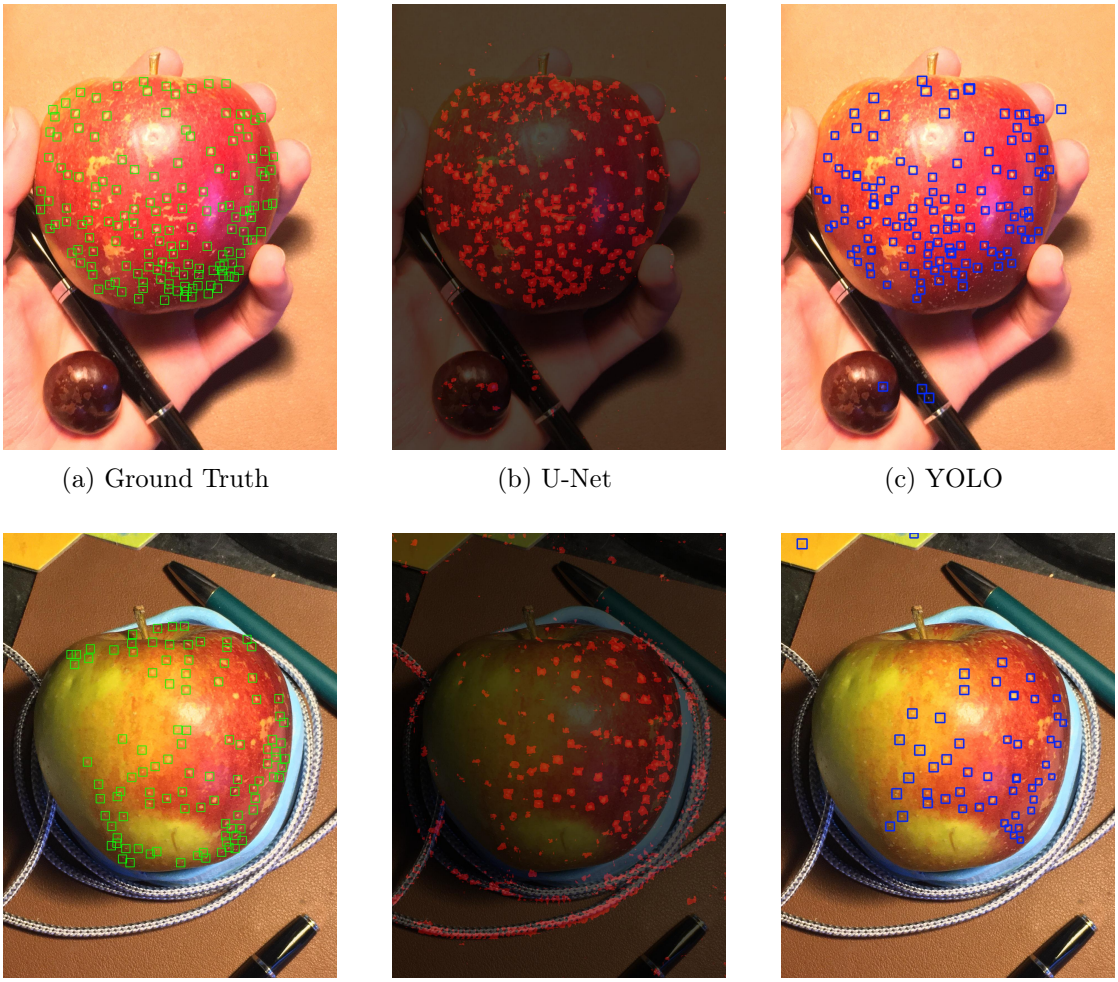
Abbildung 5.15: Wellant aus gleicher Position mit besser erkennbaren Korkporen

den, dass zumindest die beiden Modelle, U-Net nur teilweise, Korkporen besser erkennen können als der Mensch.

5.5 Hintergrund

Der Hintergrund eines Bildes kann erheblich variieren und unterschiedliche Strukturen, Farben und Objekte enthalten. In dieser Kategorie wird getestet, wie robust das Modell gegenüber unterschiedlichen und potenziell störenden Hintergründen ist. Ziel ist es zu beurteilen, ob das Modell in der Lage ist, die Korkporen von möglichen Störfaktoren im Hintergrund zu unterscheiden.

Die Modelle wurden vor zwei verschiedenen Hintergründen getestet. Abbildung 5.16(a)-(c) zeigen einen Apfel der neben einem Stift und einer Kastanie in der Hand gehalten wird und Abbildung 5.16(d)-(f) zeigen denselben Apfel aus einer Perspektive liegend auf einem Schreibtisch, umgeben von verschiedenen anderen Objekten. Beispielhaft für die Ergebnisse aller Modelle werden hier U-Net und YOLO gezeigt.



(a) Ground Truth (b) U-Net (c) YOLO

(d) Ground Truth (e) U-Net (f) YOLO

Abbildung 5.16: Apfel mit Hintergrund

U-Net weist in beiden Szenarien viele False Positives auf. In Abbildung 5.16(b) finden sich viele auf dem Apfel und in Abbildung 5.16(e) vor allem viele neben dem Apfel, speziell das Kabel ist da hervorzuheben. Abgesehen davon macht U-Net in beiden Bilden neben vielen True Positives eben auch Detektionen, die sich nicht auf dem Apfel befinden. Auch YOLO und alle anderen Modelle erkennen Korkporen an anderen Objekten oder Stellen. Das ist auf die mangelnde Diversität des Trainingsdatensatzes zurückzuführen. Die Modelle wurden nicht mit Hintergrundbildern trainiert und können so auch nicht gelernt haben, wie sie eine Korkpore von anderen Punkten unterscheiden können. YOLO ist hier noch hervorzuheben, da dieses Modell zwar auch False Positives aufweist, allerdings, wie

auch in bisherigen Ergebnissen, nur sehr wenig, wohingegen die anderen Modelle auch außerhalb des Apfels viele False Positives erkannt haben.

5.6 Defekte

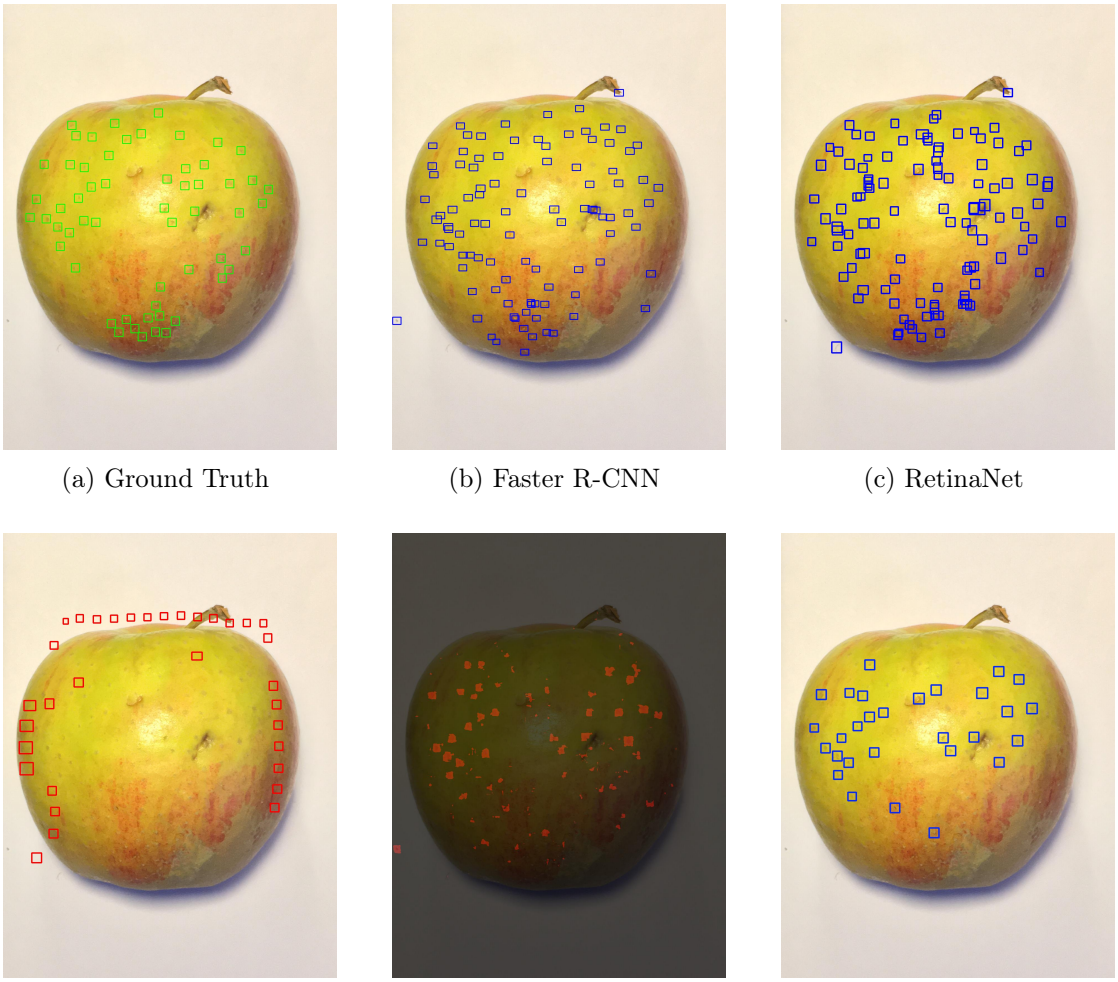
Viele Äpfel weisen Mängel oder Druckstellen auf. Dies kann durch viele Faktoren verursacht werden. In diesem Abschnitt wird überprüft, inwieweit die Modelle in der Lage sind, mit solchen Defekten auf einem Apfel umzugehen und diese nicht fälschlicherweise als Korkporen zu identifizieren. Dadurch wird sichergestellt, dass die Modelle auch mit anderen Stellen auf einem Apfel korrekt arbeiten.

Da hier ausnahmslos der Umgang mit defekten Stellen hervorgehoben werden soll, wird im Folgenden nur darauf eingegangen. Es sei aber noch angemerkt, dass die bisherigen Erkenntnisse auch hier erneut geschlossen werden können.

Jedes Modell erkennt in den defekten Stellen mindestens eine Korkpore. Faster R-CNN schafft es allerdings als einziges Modell, den linken Defekt in der Mitte des Apfels nicht als Korkpore zu identifizieren. Der andere Defekt wird jedoch von alle Modellen als solche erkannt. Wie auch schon in Abschnitt 5.5 lässt sich dies durch fehlende Trainingsbilder erklären. Da ein Defekt, ebenso wie eine Korkpore, ein Merkmal auf der Oberfläche eines Apfels ist, ist es verständlich, dass dieser auch als eine identifiziert wird.

5.7 Oben/Unten

Die Ober- und Unterseite eines Apfels unterscheiden sich deutlich von der Struktur des restlichen Apfels. Auch die Korkporen verhalten sich an diesen Stellen meist anders. So sind sie dort kleiner und häufiger. Da die Modelle nicht mit solchen Bildern gelernt haben, können diese Stellen für sie eine Herausforderung darstellen. In dieser Kategorie soll getestet werden, inwieweit die Modelle die Korkporen bei solchen abweichenden Strukturen noch gut erkennen können.



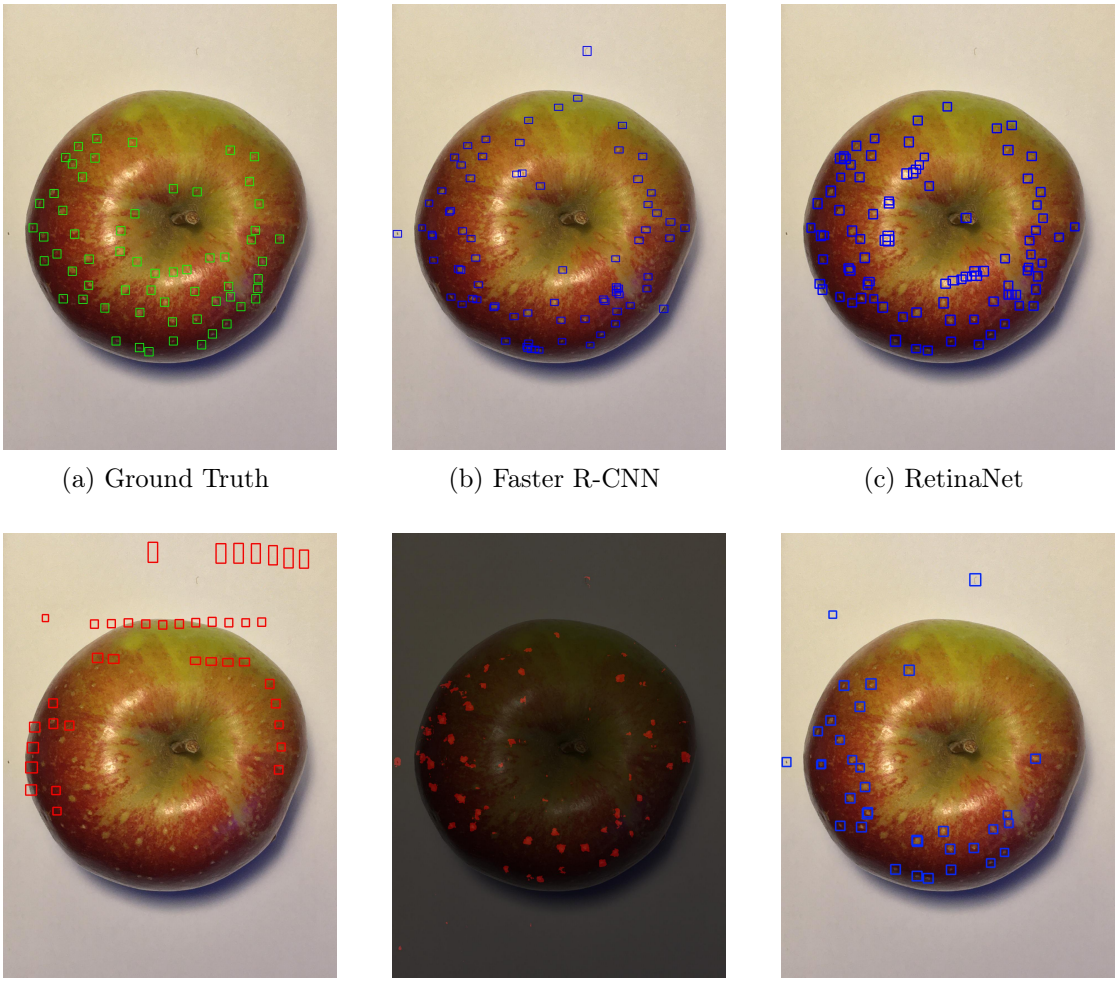
(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.17: Defekte

5.7.1 Oben

Alle Modelle erkennen die äußeren Korkporen, was daraus zu erklären ist, dass diese noch größer sind und denen mehr ähneln, die im Trainingsdatensatz hauptsächlich vertreten waren. Je mehr sich dem Stiel des Apfels genähert wird, desto weniger Detektionen treten aber auf. U-Net und RetinaNet haben als einzige noch ein paar True Positives, grundsätzlich lässt sich aber auch hier die Erkenntnis aus Abschnitt 5.5 ziehen.



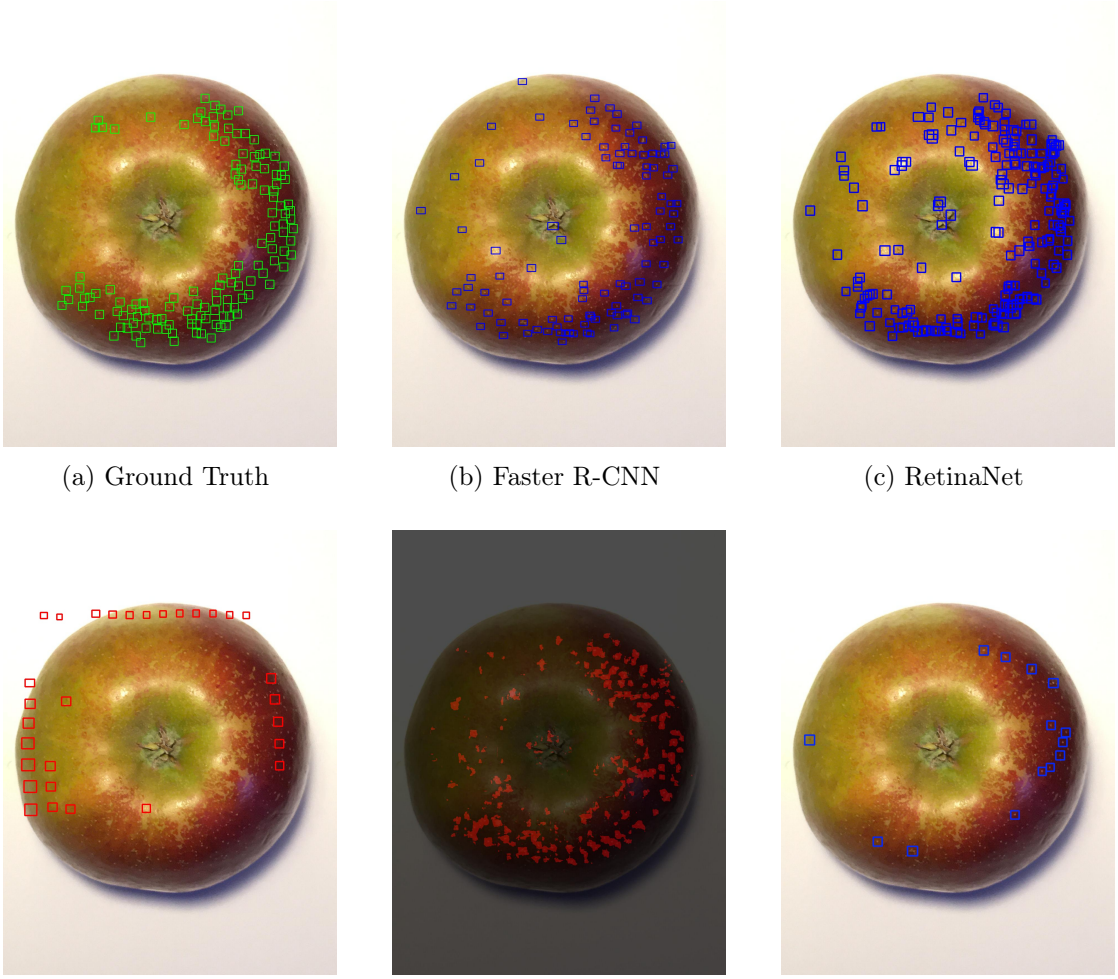
(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.18: Oben

5.7.2 Unten

Auf der Unterseite des Apfels sind die Ergebnisse zumindest bei einigen Modellen besser. Auch hier sind U-Net und RetinaNet hervorzuheben. Zwar mit vielen False Positives, aber auch mit mehr True Positives als YOLO und Faster R-CNN. Die Korkporen auf der Unterseite des Apfels sind im Trainingsdatensatz häufiger vertreten. Dies erklärt die besseren Ergebnisse. Allerdings wird auch der Blütenrest teilweise als Korkpore erkannt. Dieser war im Trainingsdatensatz jedoch nicht vorhanden, was wiederum die Schlussfolgerung aus Abschnitt 5.5 zulässt.



(a) Ground Truth (b) Faster R-CNN (c) RetinaNet

(d) SSD (e) U-Net (f) YOLO

Abbildung 5.19: Unten

6 Fazit

6.1 Zusammenfassung

In dieser Arbeit wurde die Erkennung von Korkporen auf Äpfeln mit verschiedenen Modellen untersucht. Die Ergebnisse zeigen, dass trotz einer begrenzten Anzahl von Trainingsbildern aussagekräftige Ergebnisse erzielt werden konnten. Um die Leistung weiter zu verbessern, werden verschiedene Modifikationen des Trainings vorgeschlagen.

Grundsätzlich kann jedes Modell durch eine Vergrößerung des Datensatzes optimiert werden. Mehr Bilder geben den Modellen ein differenzierteres Bild von der Struktur und den Eigenschaften der Korkporen. Konkret zeigten sich bei der Auswertung der Ergebnisse einige Schwächen aller Modelle. So wurde bei Unterunterabschnitt 5.2.2 und Unterunterabschnitt 5.2.2 deutlich, dass durch das Training von zwei Sorten kein ausreichend breites Spektrum an Apfelsorten und Korkporen im Trainingsdatensatz vertreten war. Alle Modelle, außer SSD, konnten jedoch eine hohe Erkennungsrate für die trainierten Sorten aufweisen, mit Ausnahme einiger Sonderfälle, auf die noch näher eingegangen wird. Daraus kann geschlossen werden, dass für diese Modelle etwa 50 bis 100 Bilder einer Sorte ausreichen, um fast alle Korkporen auf einfachen Bildern zu erkennen. Wie in Unterabschnitt 5.1.1 festgestellt, benötigt SSD einfach mehr Bilder, um eine bessere Erkennungsrate zu erreichen.

Abgesehen von den Problemen mit den verschiedenen Sorten hatten viele Modelle Schwierigkeiten mit anderen Faktoren auf den Bildern. Das Problem der Erkennung an den Rändern der Äpfel zog sich durch alle Ergebnisse. Mehr Bilder, die speziell diese Bereiche besser trainieren, können auch hier Abhilfe schaffen. Weitere kritische Bereiche, die im verwendeten Datensatz nicht vorkamen, waren der Ort, an dem das Licht reflektiert wurde, der obere und untere Teil des Apfels und Defekte auf den Äpfeln. All diese Faktoren sind für ein erneutes Training wesentlich, um bessere Ergebnisse zu erzielen

und ein insgesamt robusteres Modell zu erhalten. Außerdem wurden alle Bilder im Trainingsdatensatz vor einem weißen Hintergrund aufgenommen. Bei Abschnitt 5.5 hatten alle Modelle Probleme, die Korkporen vom Hintergrund zu unterscheiden. Daher sollten auch Bilder mit verschiedenen Hintergründen oder nur mit Hintergrund in den Datensatz aufgenommen werden. Abschließend bleibt zu erwähnen, dass für den Vergleich der Modelle nicht nach den optimalen Einstellungen für das Modell gesucht wurde und daher auch hier noch Verbesserungen möglich sind.

Darüber hinaus wurde die Notwendigkeit einer Verbesserung der Labelqualität festgestellt, insbesondere in Bezug auf die Variation der Korkporengröße. Vor allem kleinere Korkporen wurden oft nicht erkannt. Außerdem lernten die Modelle, dass Korkporen immer eine feste Größe haben. Nur U-Net erkannte Korkporen unabhängig von ihrer Größe. Die Bounding Boxes der anderen Modelle waren immer gleich groß, was auf die Labels im Datensatz zurückgeführt werden kann. Unterschiedlich große Labels machen das Modell robuster gegenüber unterschiedlich großen Korkporen.

Die Analyse der Versuche zeigt auch, dass alle Modelle Schwierigkeiten mit Gausschem- und Salz- und Pfeffer-Rauschen hatten. Auch hier muss in weiteren Versuchen Abhilfe geschaffen werden. Eine Möglichkeit wäre, wie bisher, den Trainingsdatensatz zu erweitern. Es ist aber auch möglich, die Bilder vor der Verarbeitung durch die Modelle einer Rauschentfernung zu unterziehen, um das Rauschen aus den Bildern zu entfernen. Da hierbei jedoch Fehler auftreten können und dieser Schritt mit mehr Aufwand und Kosten verbunden ist, ist es sinnvoller, die Modelle robust gegenüber Rauschen zu trainieren.

Wie sich bei der Auswertung der Ergebnisse zeigte, hatte RetinaNet meist viele False Positives. Dies ist ein Hinweis auf mögliches Overfitting. Abgesehen davon erkennt RetinaNet einige Korkporen, hat aber auch die bereits erwähnten Probleme. Auch Faster R-CNN erkennt einige, aber eben nicht alle Korkporen und liefert damit gute Ergebnisse.

YOLO ist dadurch aufgefallen, dass es kaum False Positives hatte. Wo andere Modelle fälschlicherweise Korkporen erkannt haben, hat YOLO selten Fehldetektionen gehabt. Das spricht für erfolgreiches Lernen der Struktur und Farbe von Korkporen. Allerdings traten häufig False Negatives auf. Ein optimiertes Trainieren kann YOLO zu einem performanten Modell machen, dass dadurch robuster gegenüber unbekannten Vorkommnissen auf den Bildern wird.

U-Net hingegen hat die meisten True Positives und erkannte über den gesamten Testdatensatz hinweg am zuverlässigsten Korkporen. Auch im Training zeigte sich durch sehr gute mAP-Werte, dass das Modell eine gute Lösung sein könnte. Dennoch hat auch U-Net teilweise einige False Positives gehabt. Das kann aber mit einem verbesserten Trainingsdatensatz behoben werden. Auffällig an den False Positives war aber auch, dass dort sehr viele kleine Detektionen waren, hinter denen sich lediglich kleine Farbveränderungen verborgen haben. Ein Ansatz wäre eine Nachbereitung der Ergebnisse. Die von U-Net kreierte Maske könnte zum Beispiel nach Größen gefiltert werden. So würden die ganz kleinen Detektionen verschwinden. Das könnte sich lohnen, da U-Net ansonsten eine gute Erkennungsrate hat.

Zusammengefasst lässt sich sagen, dass YOLO und U-Net die besten Ergebnisse aufgewiesen haben. Beide Modelle benötigen ein wenig Verbesserung, liefern aber die vielversprechendste Performance.

6.2 Einordnung (Nachfolgearbeiten)

Aufbauend auf den Ergebnissen dieser Arbeit könnten verschiedene Optimierungen im Training vorgenommen werden. Als erstes sei ein optimiertes Training angeführt. Das würde für robustere Modelle sorgen und allgemein die Erkennungsrate noch steigern.

Darüber hinaus kann auf diesen Modellen aufgebaut werden, indem die erkannten Korkporen mit ihrer Positionen gespeichert werden. So können die Korkporen in Beziehung zueinander gestellt werden. Daraus kann sich dann ein Netz aus Korkporen ergeben, mit dem zum Beispiel ein 3D-Modell erstellt werden kann. So würde sich zu jedem Apfel ein digitales Abbild der Korkporen ergeben.

Mit diesem lässt sich dann eine neue Detektion von Korkporen und deren Anordnung zueinander mit diesen digitalen Abbildern vergleichen. So wird eine neue Detektion einem Apfel zugeordnet werden.

Literaturverzeichnis

[Azgomi und Haredasht 2022] Azgomi, Hossein ; Haredasht, Fatemeh: Diagnosis of some apple fruit diseases by using image processing and artificial neural network. In: *Food Control* 145 (2022), 11, S. 109484

[Boncelet 2009] Boncelet, Charles: Image noise models. In: *The essential guide to image processing*. Elsevier, 2009, S. 143–167

[Fadnavis 2014] Fadnavis, Shreyas: Image interpolation techniques in digital image processing: an overview. In: *International Journal of Engineering Research and Applications* 4 (2014), Nr. 10, S. 70–73

[Girshick u. a. 2014] Girshick, Ross ; Donahue, Jeff ; Darrell, Trevor ; Malik, Jitendra: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, S. 580–587

[Guo-zhen u. a. 2023] Guo-zhen, DU ; Ming-xing, LU ; Ze-xu, JI ; Ji-chao, LIU: Research on apple defect detection method based on improved CNN. In: *Food and Machinery* 39 (2023), Nr. 6, S. 155–160

[Han 2013] Han, Dianyuan: Comparison of Commonly Used Image Interpolation Methods. In: *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, Atlantis Press, 2013, S. 1556–1559. – URL https://doi.org/10.2991/iccsee.2013.391. – ISBN 978-90-78677-61-1

[Hegde u. a. 2019] Hegde, Roopa B. ; Prasad, Keerthana ; Hebbar, Harishchandra ; Singh, Brij Mohan K.: Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. In: *Biocybernetics and Biomedical Engineering* 39 (2019), Nr. 2, S. 382–392

[Huang u. a. 2017] Huang, Jonathan ; Rathod, Vivek ; Sun, Chen ; Zhu, Menglong ; Korattikara, Anoop ; Fathi, Alireza ; Fischer, Ian ; Wojna, Zbigniew ; Song, Yang ; Guadarrama, Sergio u. a.: Speed/accuracy trade-offs for modern convolutional object detectors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 7310–7311

[Kaspers 2011] Kaspers, Anne: *Blob detection*, UMC Utrecht, Diplomarbeit, 2011

[Khanal u. a. 2020] Khanal, Bishnu P. ; Si, Yiru ; Knoche, Moritz: Lenticels and apple fruit transpiration. In: *Postharvest Biology and Technology* 167 (2020), sep. – Funding information: This work was funded in part by grants from the Deutsche Forschungsgemeinschaft (KN 402). We thank Dr. Erik Schulte and Mr. Stefan Eschke, Federal Plant Variety Office for samples of the apple fruit, Dr. Sandy Lang for his thoughtful comments on this manuscript.. – ISSN 0925-5214

[Khattak u. a. 2021] Khattak, Asad ; Asghar, Muhammad ; Batool, Ulfat ; Zubair, Mohd ; Ullah, Hayat ; Alrakhami, Mabrook ; Gumaei, Abdu: Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model. In: *IEEE Access* 9 (2021), 01, S. 112942–112954

[Kleefeld u. a. 2007] Kleefeld, Klaus-Dieter ; Burggraaff, Peter ; Lange, Beate: Länderübergreifende Kulturlandschaftsanalyse Altes Land. In: *Regional Cultural Plans Niedersachsen* (2007)

[Krapf 1961] Krapf, Bernhard: *Entwicklung und Bau der Lentizellen des Apfels und ihre Bedeutung für die Lagerung*, ETH Zurich, Dissertation, 1961

[Krizhevsky u. a. 2012] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems* 25 (2012)

[Lin u. a. 2017] Lin, Tsung-Yi ; Goyal, Priya ; Girshick, Ross ; He, Kaiming ; Dollár, Piotr: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*, 2017, S. 2980–2988

[Liu u. a. 2016] Liu, Wei ; Anguelov, Dragomir ; Erhan, Dumitru ; Szegedy, Christian ; Reed, Scott ; Fu, Cheng-Yang ; Berg, Alexander C.: Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* Springer (Veranst.), 2016, S. 21–37

[Mohana und Prabhakar 2015] Mohana, S.H. ; Prabhakar, C.: Automatic Detection of Surface Defects on Citrus Fruit based on Computer Vision Techniques. In: *International Journal of Image, Graphics and Signal Processing* 7 (2015), 08, S. 11–19

[Özaydın u. a. 2019] Özaydin, Umut ; Georgiou, Theodoros ; Lew, Michael: A comparison of cnn and classic features for image retrieval. In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)* IEEE (Veranst.), 2019, S. 1–4

[Patel und Mistree 2013] Patel, Vaishali ; Mistree, Kinjal: A review on different image interpolation techniques for image enhancement. In: *International Journal of Emerging Technology and Advanced Engineering* 3 (2013), Nr. 12, S. 129–133

[Redmon u. a. 2016] Redmon, Joseph ; Divvala, Santosh ; Girshick, Ross ; Farhadi, Ali: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 779–788

[Rehkugler und Throopmann 1989] Rehkugler, Gerald E. ; Throopmann, James A.: Image processing algorithm for apple defect detection. In: *Transactions of the ASAE* 32 (1989), Nr. 1, S. 267–0272

[Ren u. a. 2015] Ren, Shaoqing ; He, Kaiming ; Girshick, Ross ; Sun, Jian: Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems* 28 (2015)

[Ronneberger u. a. 2015] Ronneberger, Olaf ; Fischer, Philipp ; Brox, Thomas: U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* Springer (Veranst.), 2015, S. 234–241

[Samajpati und Degadwala 2015] Samajpati, Bhavini J. ; Degadwala, Sheshang: A Survey on Apple Fruit Diseases Detection and Classification. In: *International Journal of Computer Applications* 130 (2015), 11, S. 25–32

[Shahinfar u. a. 2020] Shahinfar, Saleh ; Meek, Paul ; Falzon, Greg: "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. In: *Ecological Informatics* 57 (2020), S. 101085

[Shermila u. a. 2023] Shermila, P. J. ; Victor, Akila ; Manoj, S. O. ; Devi, E. A.: [Detección y clasificación automática de enfermedades en frutas y hojas de cítricos utilizando un modelo basado en CNN personalizado. In: *Boletín Latinoamericano y del Caribe de Plantas Medicinales y Aromáticas* 23 (2023), dic., Nr. 2, S. 180–198. – URL https://blacpma.ms-editions.cl/index.php/blacpma/article/view/409

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
PyCharm	Entwicklungsumgebung zum Implementieren jeglichen Codes
ChatGPT	KI-Tool als Hilfestellung bei Vertständnisfragen und zur Unterstützung beim Coden
Arch Linux	Gewählte Linuxdistribution zum Arbeiten
LaTeX	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments
DeepL	KI-Tool als Hilfestellung zur Formulierung von Texten

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original