

Projektarbeit: Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

Prüfungsleistung für das Modul „Deep Learning für Computer Vision“

im Wintersemester 2022/23

des Master-Studiengangs

Informatik

an der Technischen Hochschule Mittelhessen

23. Januar 2023

Bearbeitet von:

Ben Riegel 5178979

Fabian Schwickert 5455436

Gutachter der Studienakademie Prof. Dr.-Ing. Seyed Eghbal Ghobadi

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Abkürzungsverzeichnis	V
1 Problemstellung, Ziel und Aufbau	1
2 Semantische Segmentierung von Bilddaten	3
3 Untersuchungsobjekt	7
3.1 PID-Regler	7
3.2 PID-Net	9
4 Untersuchungsumgebung	19
4.1 Datensätze für Semantische Segmentierung	19
4.2 Der SUIM-Datensatz	20
4.3 Data Augmentation	23
5 Untersuchungsergebnisse	25
Literaturverzeichnis	VI

Abbildungsverzeichnis

Abb. 1: Drei Varianten der Segmentierung	6
Abb. 2: Die PID-Net Architektur	10
Abb. 3: Overshooting bei PI-Reglern (links) und PI-Nets (rechts)	11
Abb. 4: Ein residueller Faltungsblock	12
Abb. 5: Der Aufbau der Pag-Komponente	14
Abb. 6: Der Aufbau der PAPPM-Komponente	15
Abb. 7: Der Aufbau der Bag- (a) bzw. Light-Bag-Komponente (b)	17
Abb. 8: Der Aufbau der S- und B-Head-Komponente	17
Abb. 9: Bilder (obere Zeile) und Labels (untere Zeile) aus dem SUIM-Datensatz	21
Abb. 10: Auftreten der Klassen im SUIM-Trainingsdatensatz	23
Abb. 11: Drei von acht verwendeten Data-Augmentierungs-Methoden	24
Abb. 12: Bewertung der PID-Net Architektur mit mIoU und Vorhersagezeit	26
Abb. 13: Ausgaben des PID-Net-S Modells nach 38 Trainingsepochen	28
Abb. 14: Overfitting auf Klassen	29

Tabellenverzeichnis

Tab. 1: Abbildung der RGB-Label-Werte auf Klassen-Label-Werte 22

Tab. 2: Trainingsergebnisse des PID-Net-S Modells auf dem SUIM-Datensatz 27

Abkürzungsverzeichnis

Bag	Boundary-attention-guided fusion module
BAS	Boundary Awareness
BCE	BinaryCross Entropy
B-Head	Boundary-Head
BN	Batch-Normalisierung
CE	Cross Entropy
CNN	Convolutional Neural Networks
Conv	Convolution
DAPPM	Deep Aggregation PPM
fps	frames per second
IoU	Intersection over Union
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netz
mIoU	Mean Intersection over Union
ms	Millisekunden
Pag	Pixel-attention-guided fusion module
PID	Proportional-Integral-Differential
PPM	Pyramid Pooling Module
RGB	Rot-Grün-Blau
SGD	Stochastischer Gradienten Abstieg
S-Head	Semantic-Head
SUIM	Semantic Segmentation of Underwater Imagery
Up	Upsampling

1 Problemstellung, Ziel und Aufbau

Computer spielen in der heutigen Gesellschaft eine große Rolle. Im privaten, gesellschaftlichen und betrieblichen Umfeld werden Computer eingesetzt, um Aufgaben zu automatisieren und dem Menschen Arbeit abzunehmen. In den letzten Jahren ist besonders die „Künstliche Intelligenz“ (KI) in den Vordergrund gerückt. Denn mit KI-Software sind Computer dazu in der Lage, Aufgaben wie z. B. das Erkennen von Objekten in Bildern besser als der Mensch zu erledigen. Ohne KI-Software sind Computer nur schwerlich dazu fähig.¹ Anfang 2023 kann auch KI-Software noch nicht alle Probleme perfekt lösen, die z. B. beim autonomen Fahren anfallen. Für diese Probleme wird noch leistungsstärkere KI-Software benötigt. Besonders für die Anwendung im autonomen Fahren ist es zudem entscheidend, dass die KI-Software zuverlässig die richtigen Ergebnisse in Echtzeit liefert. Denn wenn die KI-Software Gefahren nicht schnell genug erkennt, kann es zu Unfällen kommen.²

Typischerweise ist KI-Software, die in Echtzeit agieren soll, weniger komplex als KI-Software, die keiner solchen Zeitbeschränkung unterliegt. Denn je komplexer die KI-Software ist, desto mehr Berechnungen müssen für die Verarbeitung von Daten durchgeführt werden und desto länger benötigt die KI-Software für den Verarbeitungsprozess. Dadurch ist Echtzeit-KI-Software allerdings meist weniger zuverlässig und ungenauer als andere KI-Software. Denn je komplexer die KI-Software ist, desto mehr variable Parameter besitzt die KI-Software und desto genauer sind ihre Berechnungen. Für Anwendungsfelder wie z. B. das autonome Fahren muss ein Kompromiss zwischen Zeiteffizienz und Effektivität gefunden werden.³

1 Vgl. Aggerwal, Charu: Neural Networks and Deep Learning, Springer International Publishing AG (Hrsg.), 2018, <https://link.springer.com/book/10.1007/978-3-319-94463-0>, abgerufen am 14.01.2023, S. 3.

2 Vgl. Kalliomäki, Roger: Real-time object detection for autonomous vehicles using deep learning, Uppsala Universität – Department of Information Technology (Hrsg.), Juni 2019, <https://uu.diva-portal.org/smash/get/diva2:1356309/FULLTEXT01.pdf>, abgerufen am 14.01.2023, S. 1f

3 Vgl. Kalliomäki, Roger: Real-time object detection for autonomous vehicles using deep learning, a. a. O., abgerufen am 14.01.2023, S. 65. Vgl. auch Gholami, Amir et al.: A Survey of Quantization Methods for Efficient Neural Network Inference, University of California, Berkeley (Hrsg.), 21. Juni 2021, <https://arxiv.org/pdf/2103.13630.pdf>, abgerufen am 14.01.2023, S. 1.

Ein besonders für das autonome Fahren relevanter Aufgabentyp ist die „Segmentierung“ von Bildern bzw. Videos. Bei einer Segmentierungsaufgabe sollen die in den Bilddaten enthaltenen Objekte erkannt und klassifiziert werden. Beim autonomen Fahren wird dadurch z. B. der Gegenverkehr, die Straße und auch Fußgänger erkannt.⁴

Um bei derartig komplexen Aufgaben einen Kompromiss zwischen Zeiteffizienz und Effektivität zu finden, wurde in den letzten Jahren verschiedene neuartige KI-Software entwickelt. Eines dieser KI-Softwaresysteme ist „PID-Net“. PID-Net ist ein „Künstliches Neuronales Netz“ (KNN), das Ende 2022 veröffentlicht wurde und Anfang 2023 zu den zeiteffizientesten und effektivsten KNN-Modellen für die Semantische Segmentierung von Bilddaten in Echtzeit zählt. Dabei ist der besagte Kompromiss zwischen Zeiteffizienz und Effektivität mit PID-Net besonders gut gelungen.⁵

Das Ziel der vorliegenden Projektarbeit ist es, die Forschungsergebnisse zu dem Anfang 2023 noch jungen PID-Net-Modell zu bestätigen. Dafür wird das PID-Net-Modell auf einem Datensatz für die Semantische Segmentierung trainiert, der nicht von den Erfindern der PID-Net-Architektur verwendet wurde. Dabei wird die Zeiteffizienz und die Effektivität des PID-Net-Modells erhoben. Die eigens erhobenen Daten werden dann mit den von den Erfindern der PID-Net-Architektur erhobenen Daten verglichen.

Für das Erreichen des besagten Ziels ist die vorliegenden Projektarbeit wie folgt gegliedert. In Kapitel 2 der vorliegenden Projektarbeit wird zunächst genauer auf die Problemstellung eingegangen, die von PID-Net gelöst werden soll. Dabei wird genauer beschrieben, was Segmentierung ist, welche Arten der Segmentierung es gibt und wie dieser Aufgabentyp sich von anderen unterscheidet.

4 Vgl. Cakir, Senay et al.: Semantic Segmentation for Autonomous Driving: Model Evaluation, Dataset Generation, Perspective Comparison, and Real-Time Capability, Hochschule Esslingen (Hrsg.), 26. Juli 2022, <https://arxiv.org/pdf/2207.12939.pdf>, abgerufen am 14.01.2023, S. 1.

5 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, Texas A&M University (Hrsg.), 10. Juni 2022, <https://arxiv.org/pdf/2206.02066v2.pdf>, abgerufen am 14.01.2023. Vgl. auch o. V.: Real-Time Semantic Segmentation on Cityscapes test, paperswithcode.com (Hrsg.), <https://paperswithcode.com/sota/real-time-semantic-segmentation-on-cityscapes>, abgerufen am 14.01.2023. Vgl. auch o. V.: Real-Time Semantic Segmentation on CamVid, paperswithcode.com (Hrsg.), <https://paperswithcode.com/sota/real-time-semantic-segmentation-on-camvid>, abgerufen am 14.01.2023.

In Kapitel 3 wird das Untersuchungsobjekt der vorliegenden Projektarbeit beschrieben. Es wird die Idee hinter der PID-Net-Architektur und ihre Umsetzung erläutert.

In Kapitel 4 wird der Untersuchungsbereich festgelegt, in dem das Untersuchungsobjekt betrachtet wird. Es wird erklärt, was einen Datensatz für die Semantische Segmentierungs-Aufgaben ausmacht und welcher Datensatz konkret für die vorliegende Projektarbeit verwendet wird.

Im Abschließenden Kapitel 5 wird kurz beschrieben, wie die Untersuchung durchgeführt wurde. Anschließend werden ausführlich die Ergebnisse der Untersuchung vorgestellt.

2 Semantische Segmentierung von Bilddaten

In dem Forschungsfeld der KI gibt es eine Reihe von Aufgabentypen. Die bekanntesten dieser Aufgabentypen sind die Assoziation, die Klassifizierung, die Regression und die Segmentierung. Nachfolgend werden diese vier Aufgabentypen kurz beschrieben.⁶

1. Assoziation

Als Assoziation wird das Herstellen einer Korrelation zwischen Objekten bezeichnet. Dadurch können Paare oder Gruppen von Objekten gebildet werden, die eine semantische Beziehung zueinander haben.⁷ Beispielsweise könnten Artikel gruppiert werden, die häufig zusammen gekauft werden.

2. Klassifizierung

Bei einer Klassifizierungs-Aufgabe werden z. B. Bilder aus einer Gesamtmenge in klar differenzierte Teilmengen aufgeteilt. Dabei werden nicht mehrere Objekte in einem Bild se-

6 Vgl. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, Hochschule für Technik und Wirtschaft des Saarlandes – Fakultät für Wirtschaftswissenschaften – Professor für Wirtschaftsinformatik (Hrsg.), Saarbrücken, 12. Mai 2018, https://www.htwsaar.de/wiwi/fakultaet-und-personen/profile/selle-stefan/dateien/kuenstliche-neuronale-netzwerke/selle2018e_kuenstliche_neuronale_netzwerke.pdf, abgerufen am 14.01.2023, S. 13ff. Vgl. auch Strecker, Stefan: Künstliche Neuronale Netze – Aufbau und Funktionsweise, in: Arbeitspapiere WI, Nr. 10/1997, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 1997, abgerufen am 14.01.2023, S. 4.

7 Vgl. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, a. a. O., abgerufen am 14.01.2023, S. 15.

parat erkannt, sondern das Bilder als Ganzes klassifiziert. Ein Beispiel für eine Klassifizierungsaufgabe ist die Einteilung von Kunden in Kreditwürdigkeits-Klassen.⁸

3. Regression

Mittels Regression kann aus Daten der Vergangenheit zukünftiges Verhalten prognostiziert werden. Dafür modellieren KI-Systeme einen funktionalen Zusammenhang. Beispielhaft für solche Prognosen ist das „Predictive Maintenance“, bei dem vorausberechnet wird, wann eine Maschine gewartet werden muss.⁹

4. Segmentierung

Bei einer Segmentierungs-Aufgabe werden z. B. Bilder aus einer Gesamtmenge in klar differenzierte Teilmengen aufgeteilt. Allerdings werden im Gegensatz zur Klassifizierung nicht die Bilder als Ganzes klassifiziert. Stattdessen werden Bilder in Segmente aufgeteilt und jedes dieser Segmente einer Klasse zugeordnet. Dabei können die Segmente in einem Bild unterschiedlichen Klassen zugeordnet werden. Segmentierung wird z. B. häufig beim autonomen Fahren eingesetzt, um relevante Objekte wie z. B. Verkehrsschilder, die Straße oder andere Verkehrsteilnehmer im Verkehr zu erkennen.¹⁰

Die vorliegende Projektarbeit befasst sich mit einer Segmentierungsaufgabe. Daher wird nachfolgend nur noch dieser Aufgabentyp betrachtet. Segmentierungs-Aufgaben treten in verschiedenen Anwendungsfeldern auf. Besonders viel Aufmerksamkeit erhält dabei die Bilderkennung. In diesem Anwendungsfeld werden KI-Systeme eingesetzt, um den Inhalt von Bildern oder Videos zu bestimmen. In der vorliegenden Projektarbeit wird ausschließlich das Anwendungsgebiet der Bildverarbeitung betrachtet.

8 Vgl. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, a. a. O., abgerufen am 14.01.2023, S. 14. Vgl. auch Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, Technische Universität Dortmund – Fakultät für Informatik (Hrsg.), 12. November 2018, https://patrec.cs.tu-dortmund.de/pubs/theses/ma_kbranden.pdf, abgerufen am 14.01.2023, S. 5.

9 Vgl. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, a. a. O., abgerufen am 14.01.2023, S. 13f.

10 Vgl. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, a. a. O., abgerufen am 14.01.2023, S. 15. Vgl. auch Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, a. a. O., abgerufen am 14.01.2023, S. 6.

Segmentierungs-Aufgaben im Anwendungsfeld der Bilderkennung werden in drei Untertypen aufgeteilt: Die „Semantische Segmentierung“, die „Instanz Segmentierung“ und die „Panoptische Segmentierung“.¹¹ Nachfolgend werden diese drei Varianten beschrieben. In Abbildung 1 werden die Unterschiede zwischen den drei Varianten hervorgehoben. Dabei werden die Klassen, denen Objekte zugeordnet werden durch verschiedene Farben repräsentiert.

1. Semantische Segmentierung

Bei der Semantischen Segmentierung wird das gesamte Bild in Segmente unterteilt. Diese Segmente werden ausschließlich nach der semantischen Zugehörigkeit zu einer Klasse klassifiziert. Es wird nicht zwischen zwei Segmenten mit der gleichen Klassenzugehörigkeit unterschieden.¹² So würden z. B. zwei überlappend dargestellte Menschen (z. B. hintereinanderstehend) nicht durch Semantische Segmentierung unterschieden werden (siehe Abbildung 1B).

2. Instanz Segmentierung

Im Unterschied zur Semantischen Segmentierung, werden bei einer Instanz Segmentierung lediglichzählbare Objekte in einem Bild erkannt. Diese Segmente werden nach der semantischen Zugehörigkeit zu einer Klasse und der abgebildeten Instanz der Klasse klassifiziert. Es wird im Unterschied zur Semantischen Segmentierung zwischen zwei Segmenten mit der gleichen Klassenzugehörigkeit unterschieden.¹³ So würden z. B. zwei überlappend dargestellte Menschen (z. B. hintereinanderstehend) durch Instanz Segmentierung unterschieden werden (siehe Abbildung 1C).

11 Vgl. Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, Combined Faculty for the Natural Sciences and Mathematics – Universität Heidelberg, https://archiv.ub.uni-heidelberg.de/volltextserver/25750/1/thesis_kirillov.pdf, abgerufen am 13.01.2023, S. 73f.

12 Vgl. Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, abgerufen am 13.01.2023, S. 73f.

13 Vgl. Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, a. a. O., abgerufen am 13.01.2023, S. 73f.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

3. Panoptische Segmentierung

Die Panoptische Segmentierung vereint die Semantische und die Instanz Segmentierung. Bei der Panoptischen Segmentierung wird das gesamte Bild in Segmente aufgeteilt. Alle Segmente werden nach der semantischen Zugehörigkeit zu einer Klasse klassifiziert. Segmente, diezählbare Objekte repräsentieren, werden zusätzlich nach der abgebildeten Instanz der Klasse klassifiziert (siehe Abbildung 1D).¹⁴

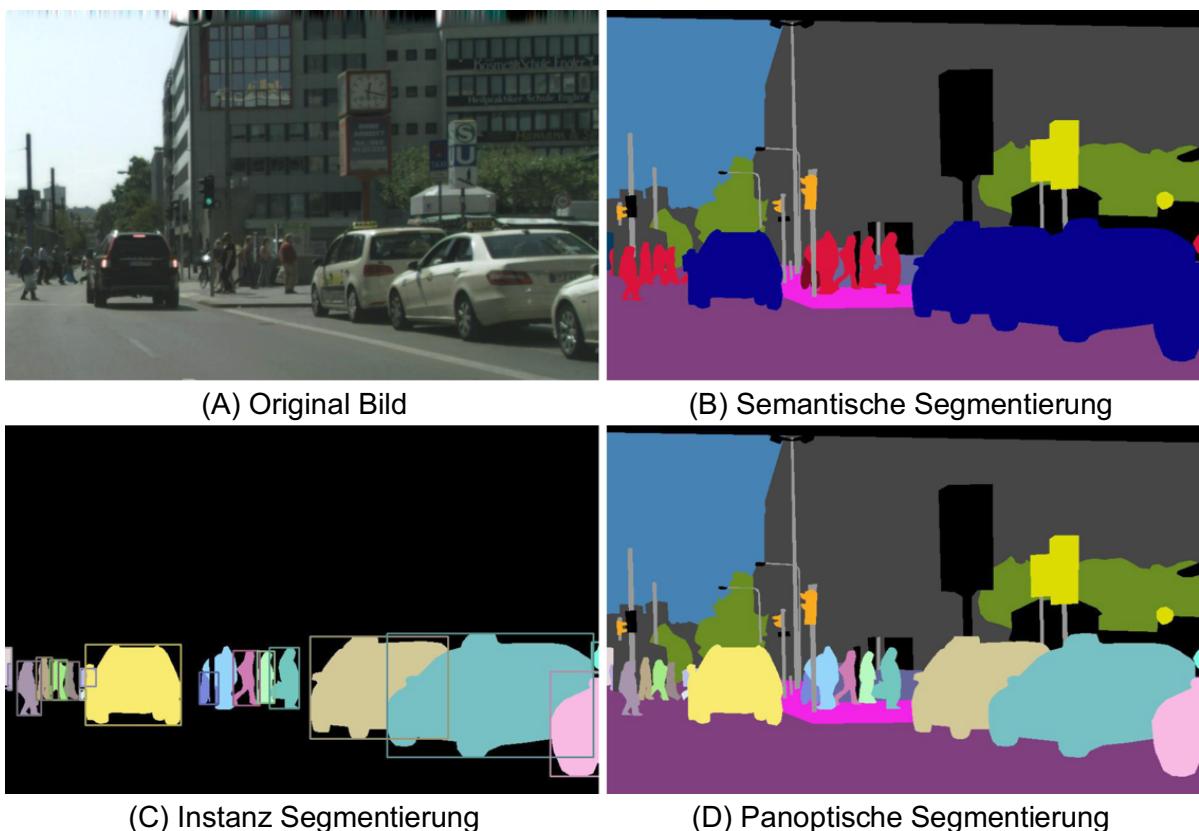


Abb. 1: Drei Varianten der Segmentierung¹⁵

In der vorliegenden Projektarbeit wird eine Semantische Segmentierungs-Aufgabe betrachtet. Daher wird nachfolgend nur noch diese Variante betrachtet.

¹⁴ Vgl. Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, a. a. O., abgerufen am 13.01.2023, S. 74.

¹⁵ Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, a. a. O., abgerufen am 13.01.2023, S. 74.

3 Untersuchungsobjekt

3.1 PID-Regler

Die PID-Net-Architektur orientiert sich an dem Aufbau eines „Proportional-Integral-Differential-Reglers“ (PID-Regler). Das Konzept des PID-Reglers stammt aus der Regelungstechnik. Die Aufgabe eines Reglers ist es, den Wert einer „Regelgröße“ möglichst auf den Wert einer „Führungsgröße“ zu bringen. Dieser angestrebte Zustand soll von dem PID-Regler auch bei sich ändernden Umgebungsbedingungen beibehalten werden. Dafür muss die Regelgröße wiederholt gemessen und mit der Führungsgröße verglichen werden. Die Differenz zwischen der Regel- und der Führungsgröße wird dann durch die Veränderung des Werts einer „Stellgröße“ iterativ reduziert.¹⁶

Das Verhalten eines PID-Reglers kann anhand einer Drohne, die auf einer bestimmten Höhe schweben soll, verdeutlicht werden. In diesem Beispiel ist die Regelgröße die Flughöhe der Drohne; die Führungsgröße ist die Ziel-Flughöhe, auf der die Drohne schweben soll und die Stellgröße ist die Rotationsgeschwindigkeit der Rotoren, die die Drohne bewegen. Damit die Drohne konstant die Ziel-Flughöhe hält, muss die Rotationsgeschwindigkeit der Rotoren wiederholt entsprechend der Differenz zwischen aktueller und Ziel-Flughöhe angepasst werden. Eine solche Anpassung ist besonders bei äußeren Einflüssen, wie z. B. einem Windstoß notwendig.

Die Besonderheit eines PID-Reglers ist, dass er drei einfacheren Regler-Komponenten vereint und dadurch die Nachteile der einzelnen Komponenten kompensiert. Ein PID-Regler besteht aus einer P-, einer I- und einer D-Komponente. Die P-Komponente bildet den funktionalen Zusammenhang zwischen der Regelgröße und der Stellgröße ab und dient der Reaktion auf aktuelle Änderungen der Regelgröße. Anhand des Drohnen-Be-

¹⁶ Vgl. O. V.: Hoehenregelung einer Kugel im Luftstrom, Hochschule Bochum – Fachbereich Elektrotechnik und Informatik – Institut für Systemtechnik – Labor für Antriebstechnik (Hrsg.), <https://www.hochschule-bochum.de/fbe/fachgebiete/institut-fuer-systemtechnik/labor-fuer-antriebstechnik/>, abgerufen am 19.01.2023, S.22.

spiels würde sich die Propeller-Geschwindigkeit proportional zu der Differenz der aktuellen und der Ziel-Flughöhe verändern.¹⁷

Das führt allerdings zu dem Problem, dass die Propellergeschwindigkeit mit zunehmender Annäherung an die Ziel-Flughöhe immer geringer wird. Auf diese Weise würde die Drohne wieder herunterfallen und die Differenz zwischen aktueller und der Ziel-Flughöhe vergrößern. Um die Drohne auf der Ziel-Flughöhe schweben zu lassen, muss die durch die Rotoren erzeugte Auftriebskraft genau der Anziehungskraft der Erde entsprechen. Die dafür notwendige Rotationsgeschwindigkeit kann allerdings auf Grund des proportionalen Zusammenhangs mit der Differenz zwischen der aktuellen und der Ziel-Flughöhe von der P-Komponente nie auf der Ziel-Flughöhe erreicht werden. Die Drohne würde unter der Ziel-Flughöhe schweben. Dies wird als „Steady-State-Fehler“ bezeichnet.¹⁸

Der Steady-State-Fehler der P-Komponente wird von der I-Komponente eines PID-Reglers durch Einbeziehen von vergangenen Daten behoben. Die I-Komponente bildet das Integral der vergangenen Differenzen zwischen gegenwärtiger und Ziel-Flughöhe. Das Integral der konstanten Differenz zwischen aktueller und Ziel-Flughöhe ist eine lineare Funktion. Die I-Komponente verändert damit die Rotationsgeschwindigkeit, bis die Differenz zwischen aktueller und Ziel-Flughöhe den Wert 0 erreicht.¹⁹

Das Problem der I-Komponente ist, dass es bei einem Steady-State-Fehler sehr nahe der Ziel-Flughöhe zu einem „Overshooting“ kommen kann. Dabei erhöht die I-Komponente die Rotationsgeschwindigkeit zu stark, sodass die Drohne über die Ziel-Flughöhe fliegt und die Differenz aus aktueller und Ziel-Flughöhe negativ wird. Dadurch wird die Rotationsgeschwindigkeit reduziert und die aktuelle Flughöhe fällt wieder unter das Ziel. Auf diese Weise kann es zu einem Pendeln um die Ziel-Flughöhe kommen.²⁰

17 Vgl. O. V.: Hoehenregelung einer Kugel im Luftstrom, a. a. O., abgerufen am 19.01.2023, S.22.

18 Vgl. Schumacher, W.: Grundlagen der Regelungstechnik, Technische Universität Braunschweig – Institut für Regelungstechnik (Hrsg.), 03. Januar 2022, https://srv.ifr.ing.tu-bs.de/static/files/lehre/vorlesungen/gdr/Skript_GdR.pdf, abgerufen am 19.01.2023, S. 150ff.

19 Vgl. O. V.: Hoehenregelung einer Kugel im Luftstrom, a. a. O., abgerufen am 19.01.2023, S.22. Vgl. auch Schumacher, W.: Grundlagen der Regelungstechnik, a. a. O., abgerufen am 19.01.2023, S. 152.

20 Vgl. Schumacher, W.: Grundlagen der Regelungstechnik, a. a. O., abgerufen am 19.01.2023, S. 168f.

Um das Overshooting der I-Komponente zu verhindern, besitzt der PID-Regler eine D-Komponente. Die D-Komponente berechnet die Ableitung der Differenz aus aktueller und Ziel-Flughöhe, um den zukünftigen Wert dieser Differenz zu erraten. Da diese Differenz zunehmend kleiner wird, hat das Ergebnis der D-Komponente immer eine negative Steigung, die die Veränderung der Rotationsgeschwindigkeit reduziert. Auf diese Weise wird bei einer zu schnellen Annäherung an die Ziel-Flughöhe und ein drohendes Overshooting die Änderung der Rotationsgeschwindigkeit gedrosselt.²¹

3.2 PID-Net

Das Untersuchungsobjekt der vorliegenden Projektarbeit ist die PID-Net-Architektur für KNNs. Diese Architektur wurde speziell für die Semantische Segmentierung von Bilddaten in Echtzeit entwickelt. Die PID-Net-Architektur verwendet vornehmlich Faltungs-Schichten, um die semantisch stark verwobenen Bilddaten effizient und effektiv zu verarbeiten. Die PID-Net-Architektur gehört daher zu den „Convolutional Neural Networks“ (CNN, zu Deutsch Neuronale Faltungsnetze).²²

Die PID-Net-Architektur spaltet ihre Datenverarbeitung entsprechend dem Konzept eines PID-Reglers in drei parallel verlaufende „Branches“ (zu Deutsch Zweige) ein. Jeder dieser Branches hat eine eigene Aufgabe, die für eine korrekte Semantische Segmentierung erledigt werden muss. Die Ergebnisse der drei Branches werden am Ende wieder zusammengeführt, um ein gemeinsames Ergebnis zu erzeugen.²³ Abbildung 2 zeigt die Struktur der PID-Net Architektur.

Der P-Branch dient der Erkennung von Detailinformationen, die sich aus den Pixeln in einem kleinen Bereich zusammensetzen. Der P-Branch versucht möglichst alle Aspekte der Detailinformationen zu erkennen und erzeugt dafür hochauflösende Featuremaps.²⁴

21 Vgl. O. V.: Hoehenregelung einer Kugel im Luftstrom, a. a. O., abgerufen am 19.01.2023, S.22. Vgl. auch Schumacher, W.: Grundlagen der Regelungstechnik, a. a. O., abgerufen am 19.01.2023, S. 168f, 172.

22 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023.

23 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3.

24 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

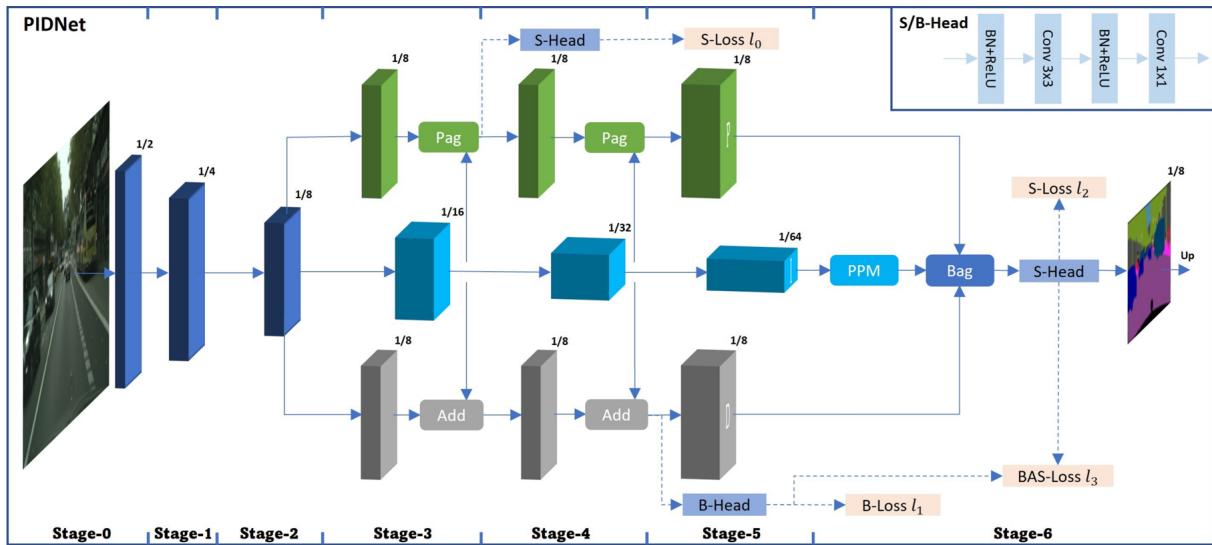


Abb. 2: Die PID-Net Architektur²⁵

Der I-Branch aggregiert lokale Detailinformationen und komplexe, globale Informationen, um semantische Verknüpfungen zwischen diesen Beiden Informationen herzustellen. Der I-Branch ermöglicht es einem PID-Net dadurch Detailinformationen in einem größeren Kontext zu betrachten.²⁶

Der D-Branch wird verwendet, um die Grenzen zwischen Objekten zu erkennen. Dadurch wird verhindert, dass die Detailinformationen aus dem P-Branch von den Kontextinformationen aus dem I-Branch überwältigt werden. Das ist das Äquivalent zu dem in Unterkapitel 3.1 betrachteten Overshooting eines PID-Reglers (siehe links in Abbildung 3). Bei einem PID-Net äußert sich ein solches Overshooting durch das Verschwimmen der Grenzen zwischen Objekten oder sogar das teilweise verschwinden eines Objekts durch Überdeckung in den Vorhersagen des Modells (siehe rechts Abbildung 3).²⁷

Die Struktur der PID-Net Architektur basiert unabhängig vom Branch auf „residuellen Faltungsblöcken“ (siehe große dreidimensional dargestellte Boxen in Abbildung 2). Diese residuellen Faltungsblöcke sind Zusammenschlüsse von mehreren Faltungsschichten je-

25 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

26 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3.

27 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 1ff.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

weils gefolgt von Aktivierung und Batch-Normalisierung. Dabei ist die Konfiguration der Faltungsschichten nicht festgelegt und variiert je nach Position in dem PID-Net Modell. Die in einen solchen Block eingehenden Daten fließen nicht nur durch diese Schicht-Folge, sondern auch über eine „Shortcut-verbindung“ direkt zum Ende des Blocks. Dort werden beide Datenflüsse addiert und als Ausgabe des Blocks zurückgegeben (siehe Abbildung 4). Die residuellen Blöcke wurden für die PID-Net Architektur gewählt, da sie effizienter sind als einfache Aneinanderreihungen von Schichten.²⁸

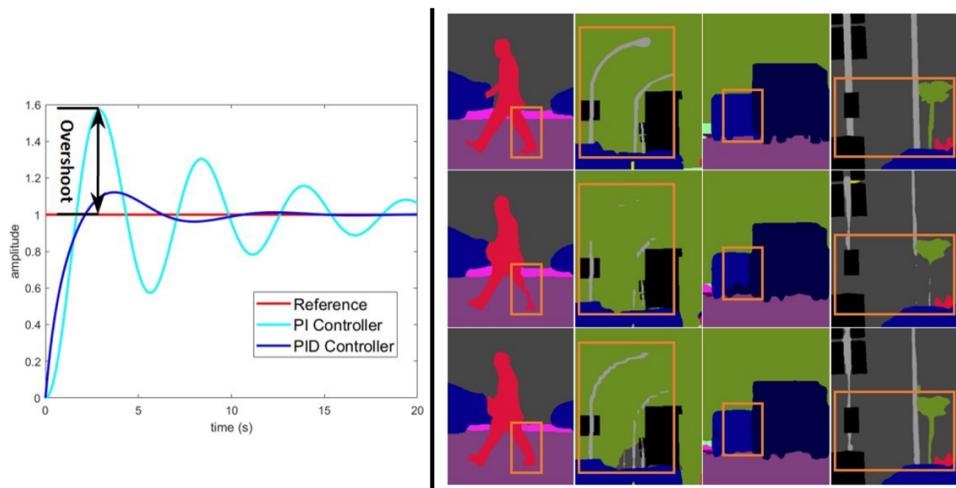


Abb. 3: Overshooting bei PI-Reglern (links) und PI-Nets (rechts)²⁹

Die PID-Net Architektur verwendet zwei Varianten von residuellen Faltungsblöcken: Eine Standard-Variante und eine Bottleneck-Variante. Die Standard-Variante besteht aus zwei Faltungsschichten inklusive Aktivierung und Batch-Normalisierung und dient ausschließlich der Erkennung von Merkmalen in Daten. Die Standard-Variante ist der Kern der Bildverarbeitung mit einem PID-Net-Modell und wird daher wesentlich häufiger als Die Bottleneck-Variante verwendet.³⁰

- 28 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3f. Vgl. auch Hong, Yuan-duo: Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, arxiv.org (Hrsg.), 01. September 2021, <https://arxiv.org/pdf/2101.06085.pdf>, abgerufen am 20.01.2023, S. 4.
- 29 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 2.
- 30 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3f. Vgl. auch Hong, Yuan-

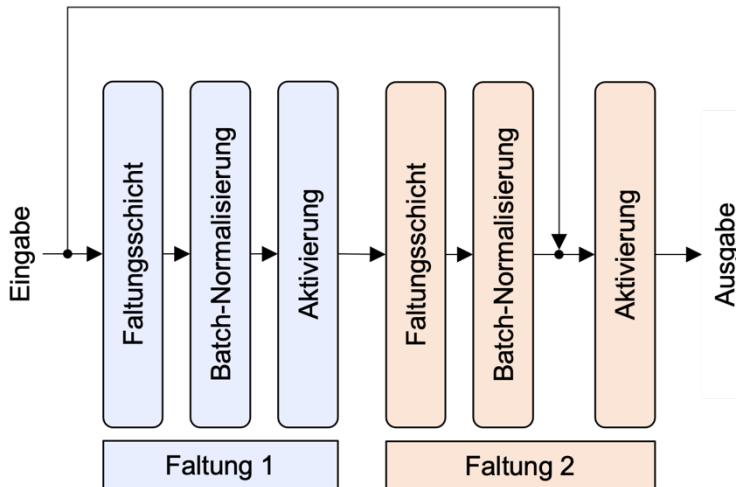


Abb. 4: Ein residueller Faltungsblock

Die Bottleneck-Variante besteht aus drei Faltungsschichten inklusive Aktivierung und Batch-Normalisierung, wobei lediglich die mittlere Faltungsschicht der Erkennung von Merkmalen in Daten dient. Die erste und die letzte Faltungsschicht dienen der effizienten Expansion der Anzahl Featuremaps. Die Bottleneck-Variante existiert nur wegen dieser Zusatzfunktion. Diese Funktion wird am Ende der drei Branches von PID-Net-Modellen verwendet, um die Dimensionen der Ausgaben der drei Branches anzugeleichen.³¹

Neben den residuellen Faltungsblöcken werden in der PID-Net Architektur noch weitere Komponenten verwendet (siehe Abbildung 2).³² Diese werden nachfolgend näher beschrieben.

duo: Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, a. a. O., abgerufen am 20.01.2023, S. 4. Vgl. auch Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/model_utils.py, github.com (Hrsg.), 27. Mai 2022, https://github.com/XuJiacong/PIDNet/blob/main/models/model_utils.py, abgerufen am 14.01.2023.

31 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 3f. Vgl. auch Hong, Yuan-duo: Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, a. a. O., abgerufen am 20.01.2023, S. 4. Vgl. auch Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/model_utils.py, a. a. O., abgerufen am 14.01.2023.

32 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4f.

1. Add-Komponente:

Die Add-Komponente ist keine Komponente in Form einer Schicht, sondern symbolisiert lediglich die elementweise Addition zweier Featuremaps. Diese Additionen werden verwendet, um die Kontextinformationen aus dem I-Branch in den D-Branch einzubringen (siehe Abbildung 2).³³

2. Pag-Komponente:

Die Pag-Komponente (Pixel-attention-guided fusion module) bringt die Kontextinformationen aus dem I-Branch in den P-Branch ein. Allerdings ist dieser Prozess wesentlich komplizierter als noch bei der Add-Komponente. Der Prozess ist komplexer, da bei einer einfachen elementweisen Addition der P-Branch mit Kontextinformationen überlastet wird. Dadurch kommt es zu dem oben angesprochenen Overshooting. Um diese Überlastung zu verhindern werden die irrelevanten Kontext-Informationen von der Pag-Komponente herausgefiltert.³⁴

Die Struktur der Pag-Komponente ist in Abbildung 5 zu sehen. Dabei steht der Operator \oplus für eine elementweise Addition und der Operator \otimes für eine elementweise Multiplikation. Die Abkürzung „Conv“ (Convolution) steht für eine Faltungsschicht und „BN“ steht für die Batch-Normalisierung. „Up“ (Upsampling) meint in Abbildung 5 das Vergrößern der Fläche einer Featuremap mittels bilinearer Interpolation. Der mit „Sum“ beschriftete Pfeil steht für das Summieren der Kanal-Werte einer Featuremap.

Das σ steht für das Ergebnis der Sigmoid-Funktion und gibt an, ob zwei Pixel zu dem gleichen Objekt gehören. Der I-Branch diese Situation auf Grund seiner umfangreichen Kontextinformationen wesentlich besser beurteilen als der P-Branch mit begrenztem Kontextwissen. Daher beeinflusst das Ergebnis des P-Branchs $(1 - \sigma)$ das Endergebnis weniger stark als das Ergebnis des I-Branchs σ .³⁵

33 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

34 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4f.

35 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4f.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

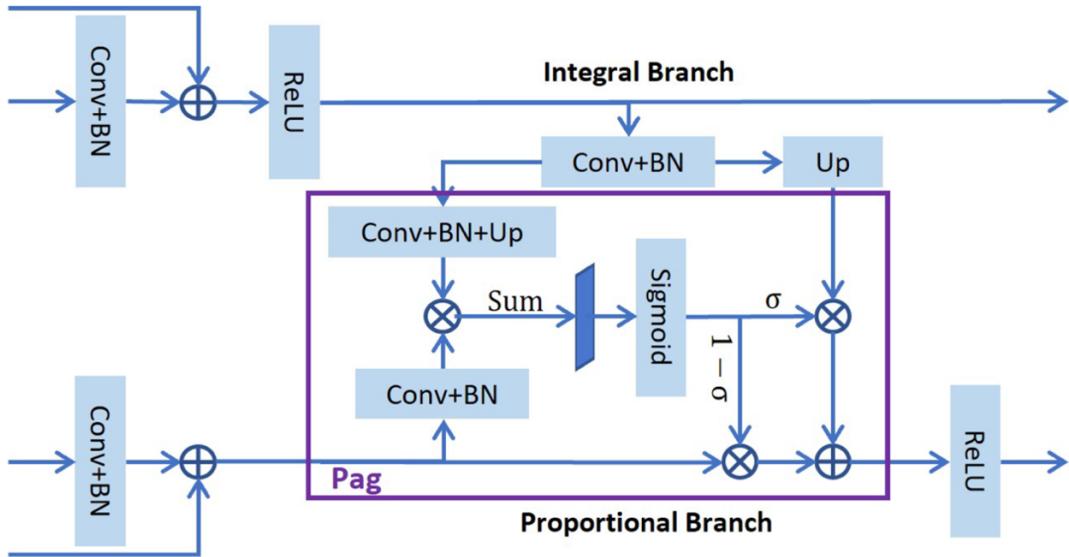


Abb. 5: Der Aufbau der Pag-Komponente³⁶

3. PPM-Komponente:

Die PPM-Komponente (Pyramid Pooling Module) sitzt am Ende des I-Branchs (siehe Abbildung 2) und dient der Verarbeitung globaler semantischer Abhängigkeiten im Bild. Diese Komponente stammt ursprünglich aus der PSPNet Architektur, wurde allerdings in mehreren anderen KNN-Modellen wiederverwendet und verbessert. Für die DDRNet Architektur wurde die effektivere DA-Variante (Deep Aggregation PPM) entwickelt, die auch in sehr großen Varianten der PID-Net Architektur verwendet werden.³⁷

Allerdings kann die DA-Variante nicht parallelisiert ausgeführt werden. Zudem enthält die DA-Variante zu viele Parameter, die die Repräsentationsfähigkeit von kleineren Modellen übersteigt. Daher kommt die DA-Variante nicht für kleinere PID-Net Modelle in Frage.³⁸ In der vorliegenden Projektarbeit wird ein solches kleineres PID-Net Modell implementiert.

36 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

37 Vgl. Hong, Yuanduo: Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, a. a. O., abgerufen am 20.01.2023, S. 4f. Vgl. auch Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

38 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

Statt der DA-Variante wird die PA-Variante (Parallel Aggregation PPM) verwendet. Dies ist eine in der Größe reduzierte und parallelisierte Form der DA-Variante und wurde für die PID-Net Architektur entwickelt. Abbildung 6 zeigt den Aufbau der PA-Variante. Dabei steht auch hier die Abkürzung „Conv“ wieder für eine Faltungsschicht und der Operator \oplus für eine elementweise Addition. Der Begriff „Upsampling“ steht für das Vergrößern der Fläche einer Featuremap mittels bilinearer Interpolation. Mittels „Concatenate“ werden die fünf auf verschiedene Weisen verarbeiteten Eingabedaten zu einer einzelnen Ausgabe vereint (siehe Abbildung 6). Dabei werden die Featuremaps lediglich übereinandergelegt. Das Ergebnis der PAPPK-Komponente hat damit so viele Kanäle wie die fünf Verarbeitungsstränge zusammen.³⁹

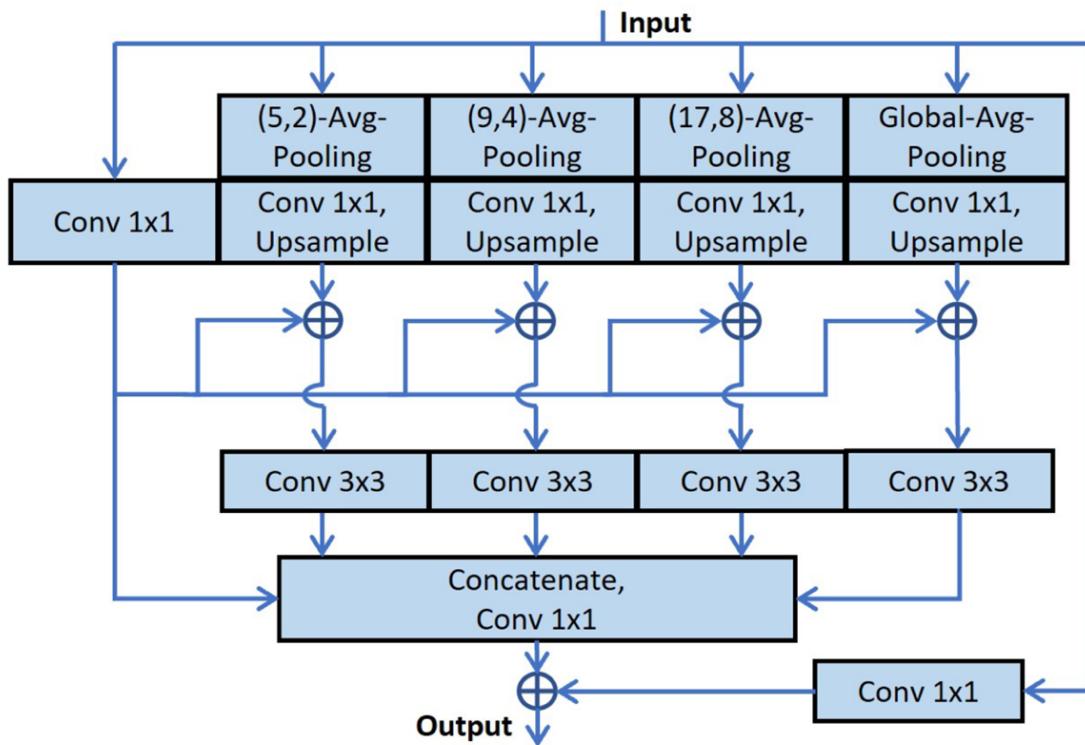


Abb. 6: Der Aufbau der PAPPK-Komponente⁴⁰

39 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5. Vgl. auch Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/model_utils.py, a. a. O., abgerufen am 14.01.2023.

40 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

4. Bag-Komponente:

Die Bag-Komponente (Boundary-attention-guided fusion module) führt die Ergebnisse der drei Branches zusammen. Entsprechend sitzt die Bag-Komponente am Ende der PID-Net Architektur. Die Bag-Komponente hat die Aufgabe die Fusion der Ergebnisse aus dem P- und dem I-Branch mit den Ergebnissen aus dem D-Branch anzuleiten, sodass es nicht zu einem Overshooting kommt (sieh Unterkapitel 3.1). Für die Grenzregionen zwischen Objekten sowie für kleine und stark detaillierte Objekte sollen die Ergebnisse des P-Bran-ches bevorzugt werden, der auf solche Aufgaben spezialisiert ist. Die anderen Regionen sollen mit den Ergebnissen des I-Banches abgedeckt werden, der den semantischen Kon-text des Bildes im Blick hat.⁴¹

Abbildung 7 zeigt die Struktur von zwei Implementierungen der Bag-Komponente. Dabei entspricht die Symbolik der Symbolik bei der Pag-Komponente in Abbildung 5. Die links in Abbildung 7 dargestellte Variante der Bag-Komponente (a) ist die Standardausführung mit einer 3x3 Faltungsschicht. Rechts (b) ist die Light-Bag Variante abgebildet, die diese 3x3 Faltungsschicht durch zwei 1x1 Faltungsschichten ersetzt. Das verändert die Funk-tion der Komponente nicht, verbessert allerdings ihre Performanz.⁴² Daher wird in der vorliegenden Projektarbeit die Light-Variante der Bag-Komponente verwendet.

In beiden Varianten der Bag-Komponente werden die Ergebnisse des P-Banches für das Endergebnis in Grenzregionen oder hochdetaillierten Bereichen verwendet; in allen an-deren Regionen wird das Ergebnis des I-Branchs verwendet. Das wird durch das Resultat σ der Sigmoid-Funktion von dem Ergebnis des D-Banches umgesetzt. Wenn der Wert von $\sigma > 0,5$ ist, werden die Ergebnisse des P-Banches verwendet; anderenfalls werden die Ergebnisse des I-Banches verwendet.⁴³

41 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

42 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

43 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

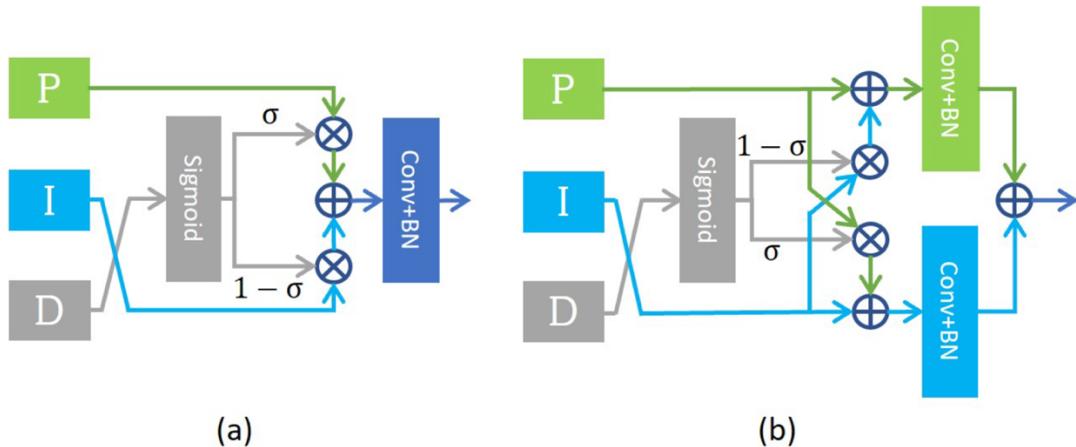


Abb. 7: Der Aufbau der Bag- (a) bzw. Light-Bag-Komponente (b)⁴⁴

5. S- und B-Head-Komponenten

Die S-Head- (Semantic-Head) und B-Head-Komponenten (Boundary-Head) sind alle gleich nach dem in Abbildung 8 gezeigten Schema aufgebaut. Sie dienen der Aufbereitung der Ausgaben einer Komponente für die Berechnung eines Loss-Werts oder die Anzeige des Endergebnisses. Entsprechend befindet sich ganz am Ende der PID-Net Architektur eine S-Head-Komponente (siehe Abbildung 2).⁴⁵

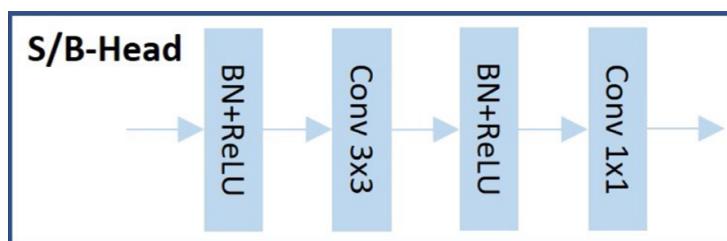


Abb. 8: Der Aufbau der S- und B-Head-Komponente⁴⁶

Es befindet sich allerdings noch eine S-Head-Komponente in Stage 4, die von der Ausgabe der ersten Pag-Komponente gespeist wird. Diese S-Head-Komponente dient der Optimierung des gesamten Modells. Die einzige B-Head-Komponente der PID-Net Architektur be-

44 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5.

45 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

46 Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

findet sich in Stage 5 und wird durch die Ausgabe der zweiten Add-Komponente gespeist (siehe Abbildung 2). Diese B-Head-Komponente soll speziell bei der Optimierung der Erkennung der Grenzregionen helfen.⁴⁷

Die Ausgaben der beschriebenen S- und B-Head-Komponenten werden für die Erreichung der Loss-Werte verwendet, mit denen PID-Net Modelle optimiert werden. Die PID-Net Architektur sieht für die verschiedenen S- bzw. B-Head-Komponenten unterschiedliche Loss-Funktionen vor. Für die Berechnung der Loss-Komponente l_0 wird die gewichtete Kreuzentropie (CE, Cross Entropy) als Loss-Funktion auf die Ausgabe der vorderen S-Head-Komponente angewendet (siehe Abbildung 2). Für die Berechnung der Loss-Komponente l_1 wird die gewichtete binäre Kreuzentropie (BCE, Binary Cross Entropy) als Loss-Funktion auf die Ausgabe der B-Head-Komponente angewendet (siehe Abbildung 2). Für die Berechnung der Loss-Komponente l_2 wird die gewichtete CE-Loss-Funktion auf die Ausgabe der hinteren S-Head-Komponente angewendet (siehe Abbildung 2).⁴⁸

Für die Berechnung der Loss-Komponente l_3 wird ein Boundary Awareness CE-Loss-Funktion (BAS-Loss in Abbildung 2) auf die Ausgabe der hinteren S-Head-Komponente angewendet. Diese BAS-Loss-Funktion berechnet die gewichtete Kreuzentropie genau wie die bei den Loss-Komponenten l_0 und l_2 . Allerdings wird durch die Ausgabe der B-Head-Komponente bestimmt, dass dies nur im näheren Umfeld von Grenzregionen und detaillierten Bereichen stattfinden soll

$$l_3 = - \sum_{i,c} \{1: b_i > t\} \cdot [s_{i,c} \cdot \log(\hat{s}_{i,c})].$$

In der Formel für die Berechnung von l_3 steht i für den Index eines Pixels und c für eine von n Klassen, denen der Pixel zugewiesen werden kann. b_i ist die Ausgabe der B-Head-Komponente, $s_{i,c}$ die Ausgabe der S-Head-Komponente und $\hat{s}_{i,c}$ die Klasse, der der Pixel

47 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

48 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4. Vgl. auch Xu, Jiacong: PIDNet: XuJiacong/PIDNet – utils/criteriom.py, github.com (Hrsg.), 28. Mai 2022, <https://github.com/XuJiacong/PIDNet/blob/main/utils/criterion.py>, abgerufen am 14.01.2023.

am Index i tatsächlich zugeordnet werden sollte (ground truth). Der Parameter t ist ein Grenzwert, der angibt, wann die Kreuzentropie berechnet werden soll. Dieser Parameter wurde von den Erfindern der PID-Net Architektur auf $t = 0,8$ festgelegt.⁴⁹

Alle benannten Loss-Komponenten werden als gewichtete Summe zu dem Gesamt-Loss aggregiert, der zurückpropagiert wird

$$Loss = \lambda_0 \cdot l_0 + \lambda_1 \cdot l_1 + \lambda_2 \cdot l_2 + \lambda_3 \cdot l_3 .$$

Dabei wurden die Gewichte λ_j durch die Erfinder der PID-Net Architektur festgelegt:

$$\lambda_0 = 0,4 \mid \lambda_1 = 20 \mid \lambda_2 = 1 \mid \lambda_3 = 1 .^{50}$$

Für die Optimierung von PID-Net Modellen wurde von den Erfindern der Architektur der Stochastische Gradienten Abstieg (SGD, Stochastic Gradient Descent) in einer stark parametrisierten Form verwendet. Es wurde ein Momentum von 0,9 und ein Weight-Decay von 0,0001 verwendet. Die Lernrate für die Optimierung eines PID-Net Modells wurde von den Entwicklern so festgelegt, dass sie während des Trainingsprozesses von 0,1 über 0,01 zu 0,001 reduziert wird.⁵¹

4 Untersuchungsumgebung

4.1 Datensätze für Semantische Segmentierung

Bilddatensätze für Semantische Segmentierung bestehen aus den Bilddaten, die segmentiert werden soll und aus „Labels“, die die korrekte Segmentierung eines Bilds angeben. Diese Labels werden gemäß einer Loss-Funktion pixelweise mit der Vorhersage eines KNNs verglichen, um einen Loss-Wert zu errechnen.⁵²

49 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

50 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4.

51 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 6.

52 Vgl. Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, a. a. O., abgerufen am 14.01.2023, S. 3f, 41f.

Damit dieser pixelweise Vergleich stattfinden kann, müssen die Labels und die Ausgaben eines KNNs das gleiche Format haben. Die Ausgaben eines PID-Net Modells haben nicht die Flächen-Größe des eingegebenen Bildes und des zugehörigen Labels. Um diese Flächen-Größe herzustellen, werden die Ausgaben eines PID-Net Modells mit bilinearer Interpolation auf diese Größe skaliert (siehe „Up“ am rechten Rand von Abbildung 2).⁵³

Die Anzahl Kanäle – die Tiefe – der Labels und der Vorhersagen eines KNNs müssen ebenso übereinstimmen. Die Ausgaben eines KNNs für Semantische Segmentierung haben immer genau so viele Kanäle, wie Klassen segmentiert werden sollen. Labels liegen häufig nicht in einer solchen Form vor. Stattdessen werden Labels oft als RGB-Bild (Rot-Grün-Blau) bereitgestellt, wobei eine Farbe eine Klasse repräsentiert (sieh Abbildung 1B). Um die Tiefe von Labels und KNN-Ausgaben anzugeleichen, werden i. d. R. die RGB-Labels in Klassen-Labels mit einer Tiefe der Anzahl zu segmentierender Klassen überführt.⁵⁴

4.2 Der SUIM-Datensatz

Die vorliegenden Projektarbeit verwendet den „Semantic Segmentation of Underwater Imagery“ (SUIM) Datensatz.⁵⁵ Der Datensatz enthält Bilder von Unterwasserlandschaften (siehe Abbildung 9) und unterscheidet zwischen den folgenden acht Klassen:

1. Hintergrund (Wasserkörper)
2. Taucher
3. Unterwasser-Flora
4. Wracks und Ruinen
5. Roboter und Maschinen
6. Riffe und Korallen

53 Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 4. Vgl. auch Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/pidnet.py, github.com (Hrsg.), 10. Juni 2022, <https://github.com/XuJiacong/PIDNet/blob/main/models/pidnet.py>, abgerufen am 23.01.2023.

54 Vgl. Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, a. a. O., abgerufen am 14.01.2023, S. 65.

55 Der SUIM-Datensatz ist unter dem folgenden Link auf Kaggle.com abrufbar: <https://www.kaggle.com/datasets/ashish2001/semantic-segmentation-of-underwater-imagery-suim>.

7. Unterwasser-Flora

8. Meeresboden und Steine

Der SUIM-Datensatz enthält ca. 1500 für eine Semantische Segmentierung gelabelte Bilder für das Training und ca. 110 für eine Semantische Segmentierung gelabelte Bilder für das Testen.⁵⁶

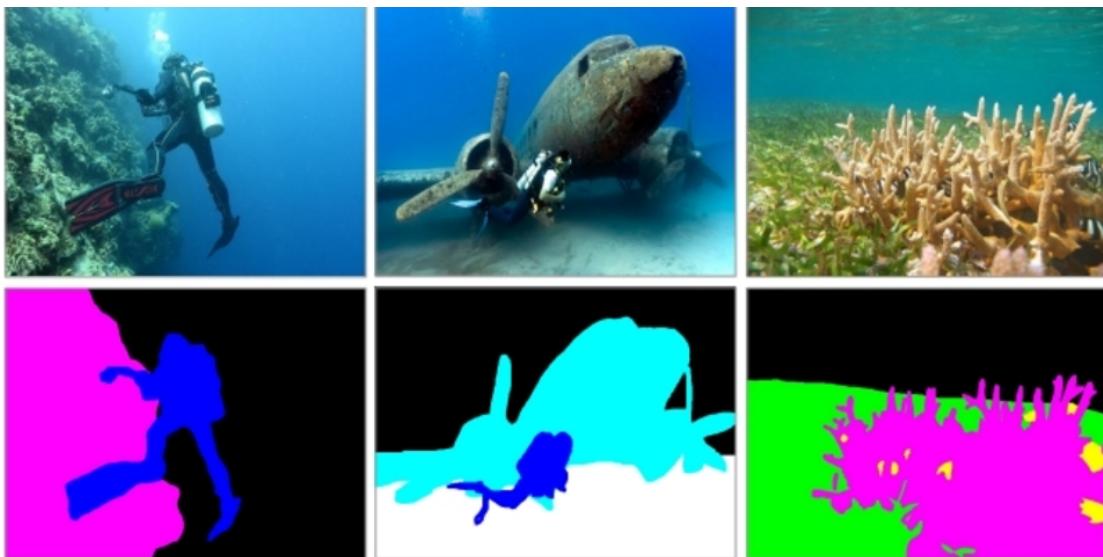


Abb. 9: Bilder (obere Zeile) und Labels (untere Zeile) aus dem SUIM-Datensatz⁵⁷

Die Bilder und zugehörigen Labels haben unterschiedliche Flächen-Größen. Für eine Verarbeitung mit einem KNN werden die Bilder alle auf die Größe 1024x1024 skaliert. Dabei wird eine geringfügige Verzerrung in Kauf genommen. Alle Bilder und auch die dazugehörigen Labels liegen mit drei Farbkanälen für den RGB-Farbraum vor. Die Bilder müssen für eine Verarbeitung mit einem KNN in der Tiefe nicht weiter verändert werden. Die Labels müssen allerdings von den drei Farbkanälen auf acht Klassen-Kanäle abgebildet werden. Dafür wird die in Tabelle 1 dargestellte Menge an Abbildungen von RGB-Farbwerten auf die oben angeführten Klassen angewendet.⁵⁸

56 Vgl. Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, arxiv.org (Hrsg.), 13. September 2020, <https://arxiv.org/abs/2004.01241>, abgerufen am 23.01.2023, S. 1f.

57 Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, a. a. =., abgerufen am 23.01.2023, S. 3.

58 Vgl. Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, a. a. O., abgerufen am 23.01.2023, S. 2.

Farbe	RGB-Code	Klassenbezeichnung	Klassen-Nr.
Schwarz	(0, 0, 0)	Hintergrund (Wasserkörper)	1
Dunkelblau	(0, 0, 255)	Taucher	2
Grün	(0, 255, 0)	Unterwasser-Flora	3
Himmelblau	(0, 255, 255)	Wracks und Ruinen	4
Rot	(255, 0, 0)	Roboter und Maschinen	5
Rosa	(255, 0, 255)	Riffe und Korallen	6
Gelb	(255, 255, 0)	Unterwasser-Flora	7
Weiß	(255, 255, 255)	Meeresboden und Steine	8

Tab. 1: Abbildung der RGB-Label-Werte auf Klassen-Label-Werte⁵⁹

Für ein erfolgreiches Training eines KNNs ist es wichtig, dass die verwendeten Daten eine hohe Qualität und Varianz haben. Bei Klassifizierungs- und Segmentierungsaufgaben ist es besonders wichtig, dass alle relevanten Klassen in ausreichender Zahl gleichmäßig in dem Datensatz enthalten sind. Andernfalls kann es passieren, dass unterrepräsentierte Klassen gar nicht erkannt werden oder ganze Bilder pauschal mit überrepräsentierten Klassen klassifiziert werden.⁶⁰

Abbildung 10 zeigt die Verteilung der Klassen in dem SUIM-Trainingsdatensatz. Dabei sind auf der X-Achse die einzelnen Klassen abgetragen. Die Y-Achse gibt die Anzahl der Bilder in dem Trainingsdatensatz an, in denen die jeweilige Klasse mindestens einmal zu sehen ist. Es fällt auf, dass der SUIM-Trainingsdatensatz stark von den Klassen „Hintergrund“, „Riffe und Korallen“ sowie „Unterwasser-Flora“ dominiert wird. Die Klasse „Meeresboden und Steine“ ist in einer einigermaßen großen Menge an Bildern vertreten. Aber

59 Vgl. Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, a. a. O., abgerufen am 23.01.2023, S. 2.

60 Vgl. Bischke, Benjamin et al.: Segmentation of Imbalanced Classes in Satellite Imagery using adaptive uncertainty weighted class loss, Technische Universität Kaiserslautern und Deutsches Forschungszentrum für Künstliche Intelligenz (Hrsg.), Deutschland, 2018, https://www.dfk1.de/fileadmin/user_upload/import/9877_IGARSS_2018_bischke_uncertainty_loss.pdf, abgerufen am 23.01.2023, S. 1. Vgl. auch Li, Zeju; Kamnitsas, Konstantinos; Glocker, Ben: Analyzing Overfitting under Class Imbalance in Neural Networks for Image Segmentation, arxiv.org (Hrsg.), 20. Februar 2021, <https://arxiv.org/pdf/2102.10365.pdf>, abgerufen am 23.01.2023.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

die Klassen „Taucher“, „Unterwasser-Flora“, „Wracks und Ruinen“ sowie „Roboter und Maschinen“ sind stark unterrepräsentiert.⁶¹

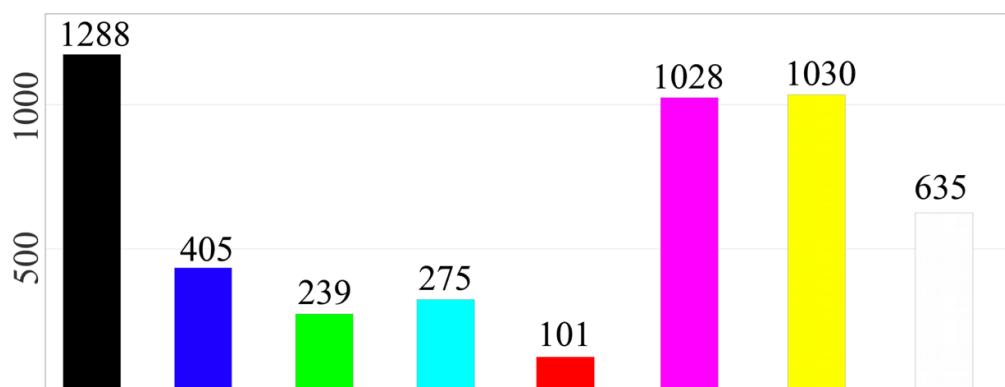


Abb. 10: Auftreten der Klassen im SUIM-Trainingsdatensatz⁶²

Zudem ist dem Projektteam während des Trainings aufgefallen, dass die Klassen „Hintergrund“, „Riffe und Korallen“ sowie „Meeresboden und Steine“ sowie sie in einem Bild enthalten sind, einen sehr großen prozentualen Anteil des Bildes Einnehmen.

4.3 Data Augmentation

Datensätze für das Training von KNNs sollten für Komplexe Problemstellung so groß wie möglich sein und eine hohe Varianz aufweisen. Bekannte Datensätze zur Semantischen Segmentierung sind Cityscapes mit 5.000 hochwertig annotierten Bildern und COCO-Stuff mit 10.000 hochwertig annotierten Bildern. Viele KNN Modelle und auch PID-Net Modelle haben in der Vergangenheit gezeigt, dass auf diesen Datensätzen gute Semantische Segmentierungs-Ergebnisse erzielt werden können.⁶³

Der in der vorliegenden Projektarbeit verwendete SUIM-Trainingsdatensatz besteht nur aus ca. 1500 Bildern. Hinzu kommt, dass die Daten eine Klassen-Imbalance haben. Um

61 Vgl. Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, a. a. O., abgerufen am 23.01.2023, S. 2.

62 Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, a. a. O., abgerufen am 23.01.2023, S. 2.

63 Vgl. Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, a. a. O., abgerufen am 14.01.2023, S. 3. Vgl. auch Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 5f.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

diese beiden Probleme auszugleichen, wurde in der vorliegenden Projektarbeit Daten-Augmentierung vorgenommen. Dabei wurden die folgenden acht Augmentierungs-Methoden separat verwendet. Abbildung 11 zeigt beispielhaft n dieser Augmentierungen.

1. Zufällige Änderung der Bildhelligkeit
2. Zufällige Änderung der Bildkontrastierung
3. Zufällige Änderung der Bildunschärfe
4. Invertierung der Bildfarbe (siehe Abbildung 11b)
5. Horizontale, vertikale und kombinierte Spiegelung
6. Erzeugen von Bildausschnitten zufälliger Größe und Position (siehe Abbildung 11d)
7. Verrauschen des Bilds
8. Zufällige Änderung der Bildfarbsättigung (siehe Abbildung 11c)

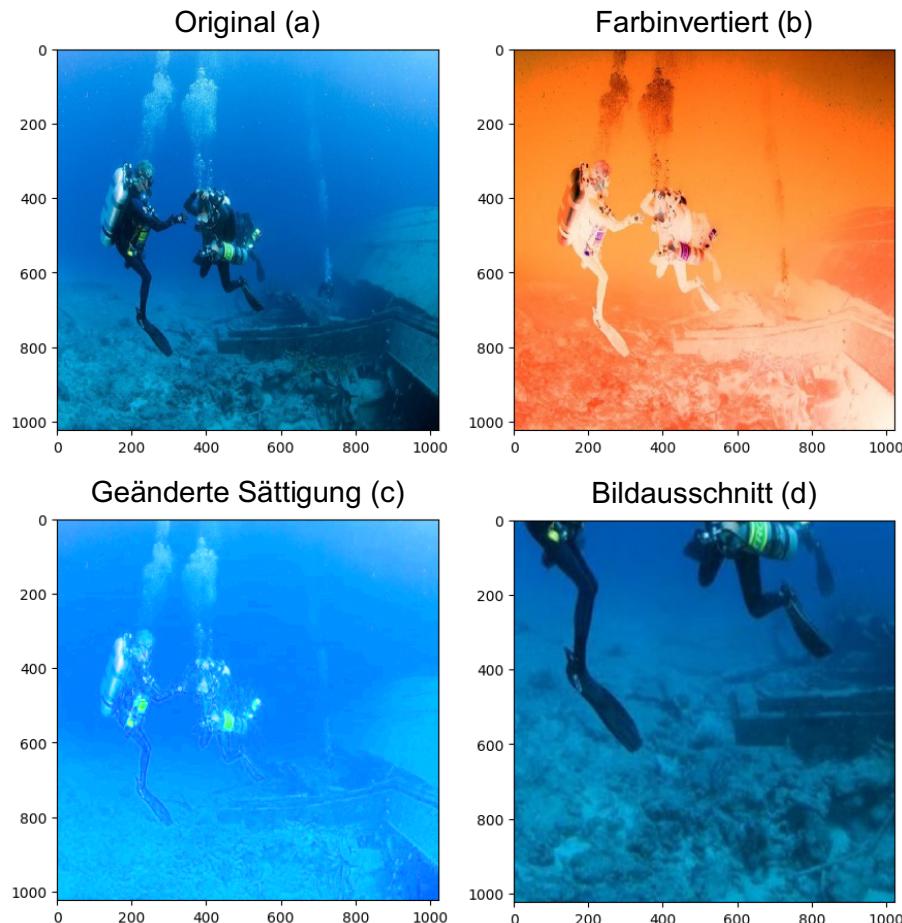


Abb. 11: Drei von acht verwendeten Data-Augmentierungs-Methoden

Alle diese Augmentierungs-Methoden auf alle Bilder angewendet. Zusätzlich wurden Bilder vervielfältigt, die unterrepräsentierte Klassen beinhalteten. Besonders das Erzeugen von Bildausschnitten von zufälliger Größe und Position wurde dafür herangezogen, überproportional viele Bilder von unterrepräsentierten Klassen mit neuem Inhalt zu schaffen.

5 Untersuchungsergebnisse

Die Untersuchungsergebnisse der vorliegenden Projektarbeit werden anhand der Metriken „Mean Intersection over Union“ und der „Vorhersagezeit“ bewertet. Diese beiden Metriken werden nachfolgend kurz beschrieben. Abbildung 12 zeigt als Vergleichswert die Ergebnisse dieser Metriken der PID-Net Modelle auf den Cityscapes Testdaten.

1. Mean Intersection over Union in %

„Mean Intersection over Union“ (mIoU) ist die zentrale Bewertungsmetrik für Segmentierungsaufgaben. Sie berechnet sich als Mittelwert aus der „Intersection over Union“ (IoU) für alle Segmentierungs-Klassen. Die IoU für eine Klasse wird als das Verhältnis von korrekt klassifizierten Pixeln (Intersection) zu der Anzahl aller Pixel (Union) berechnet.⁶⁴

Das PID-Net-S Modell wurde bereits auf dem Cityscapes Datensatz trainiert und hat dabei eine mIoU von 78,6% erreicht (siehe Abbildung 12).⁶⁵ Die vorliegende Projektarbeit versucht dieses Ergebnis auf dem SUIM-Datensatz zu validieren. Allerdings lässt sich schon im Voraus sagen, dass dies vermutlich nicht möglich sein wird. Denn der SUIM-Datensatz ist wesentlich kleiner als der Cityscapes-Datensatz und hat eine große Imbalance der Klassen. Das Ziel der vorliegenden Projektarbeit wurde daher erst einmal optimistisch auf eine mIoU von 50% festgelegt.

2. Vorhersagezeit in Millisekunden

Die Vorliegende Projektarbeit befasst sich mit Semantischer Segmentierung von Bilddaten in Echtzeit. Dieser Thematik ist inhärent, dass die Vorhersagezeit des KNNs gering

⁶⁴ Vgl. Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, a. a. O., abgerufen am 14.01.2023, S. 62.

⁶⁵ Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 8.

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

sein muss. Damit ein Standard-Video mit 30 frames per second (fps) von einem KNN segmentiert werden kann, muss jedes Bild in ca. 34 Millisekunden (ms) verarbeitet werden. Ab diesem Zeitpunkt kann von Echtzeitfähigkeit gesprochen werden.

Bei dem Training des PID-Net-S Modells auf dem Cityscapes Datensatz konnte eine Vorhersagegeschwindigkeit von 93,2 fps erzielt werden (siehe Abbildung 12).⁶⁶ Das entspricht einer Vorhersagezeit von ca. 11 ms pro Bild. Die vorliegende Projektarbeit versucht dieses Ergebnis auf dem SUIM-Datensatz zu validieren. Im Gegensatz zur mIoU, kann dieses Ergebnis in der vorliegenden Projektarbeit bestätigt werden. Denn die PID-Net Architektur wird nicht verändert. Das Ziel der vorliegenden Projektarbeit ist daher eine Vorhersagezeit von 11 ms pro Bild.

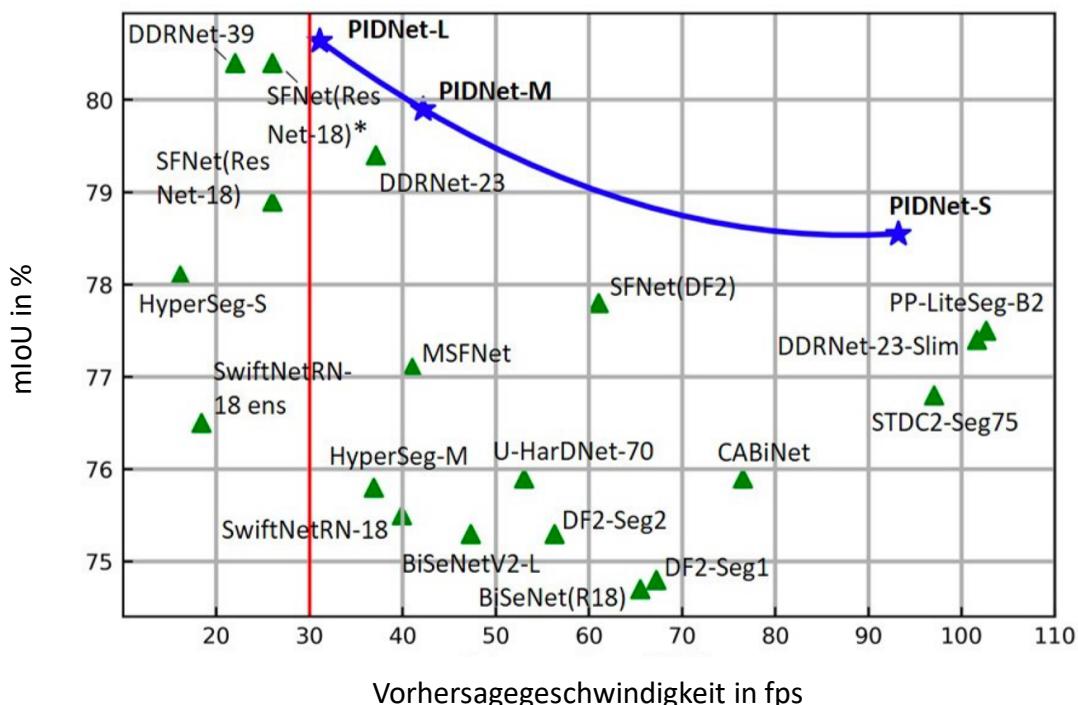


Abb. 12: Bewertung der PID-Net Architektur mit mIoU und Vorhersagezeit⁶⁷

Für die vorliegende Projektarbeit wurde ein PID-Net-S Modell auf dem SUIM-Datensatz für 38 Epochen trainiert. Dabei konnte durchweg bestätigt werden, dass dieses Modell

⁶⁶ Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 8.

⁶⁷ Vgl. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller, a. a. O., abgerufen am 14.01.2023, S. 1.

eine Vorhersagezeit von ca. 11 ms pro Bild hat. Die Vorhersagegenauigkeit dieses Modells konnte allerdings wie erwartet nicht bestätigt werden. Nach zehn Epochen konnte eine mIoU von ca. 9%, nach 20 Epochen eine mIoU von ca. 12%, nach 30 Epochen eine mIoU von ca. 21% und nach 38 Epochen eine mIoU von ca. 28% festgestellt werden (siehe Tabelle 2).

Dabei zeigte sich, dass die mIoU auf den Trainingsdaten immer zwischen 6% - 8% höher lag als bei den besagten Ergebnissen auf den Testdaten. Wenn überhaupt kann hier von einem minimalen Overfitting auf den Trainingsdaten gesprochen werden.

Epoch	mIoU in %	Vorhersagezeit in ms
10	9	11
20	12	11
30	21	11
38	28	11

Tab. 2: Trainingsergebnisse des PID-Net-S Modells auf dem SUIM-Datensatz

Abbildung 13 zeigt drei Bilder aus dem SUIM-Testdatensatz (obere Zeile) und die von dem 38 Epochen trainierten PID-Net-S Modell produzierten Ausgaben zu dem jeweiligen Bild als Maske über dem Eingabe-Bild (untere Zeile). Bei der Betrachtung dieser Ausgaben fällt auf, dass es große Unterschiede zu optimalen Labels aus Abbildung 9 gibt.

Es ist oft eine Menge Rauschen zu sehen, was sich in dem linken und dem rechten Bild in Abbildung 13 vor allem an den Grenzregionen abspielt. Häufig werden Objekte falsch klassifiziert (siehe das Schiffswrack im linken Bild in Abbildung 13) und die Objekte werden auch nicht scharf in den Grenzregionen abgegrenzt (siehe die Fische im mittleren Bild in Abbildung 13). Und in einigen Fällen werden Bestandteile eines Objekts, weit vom Zentrum des Objekts entfernt sind, nicht mehr zu diesem Objekt hinzugezählt (siehe die Flügel des Flugzeugwracks im rechten Bild in Abbildung 13).

Auf der anderen Seite kann die rudimentäre Erkennung von Objekten (siehe das Schiffswrack, den Tauscher und den Hintergrund im linken Bild in Abbildung 13; siehe das Riff und den Meeresboden im mittleren Bild in Abbildung 13; siehe das Flugzeugwrack und den Meeresboden im rechten Bild in Abbildung 13) als Erfolg vermerkt werden. Diese

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

Fähigkeit des eigens trainierten PID-Net-S Modells zieht durch die Ausgaben für die SUIM-Testdaten.

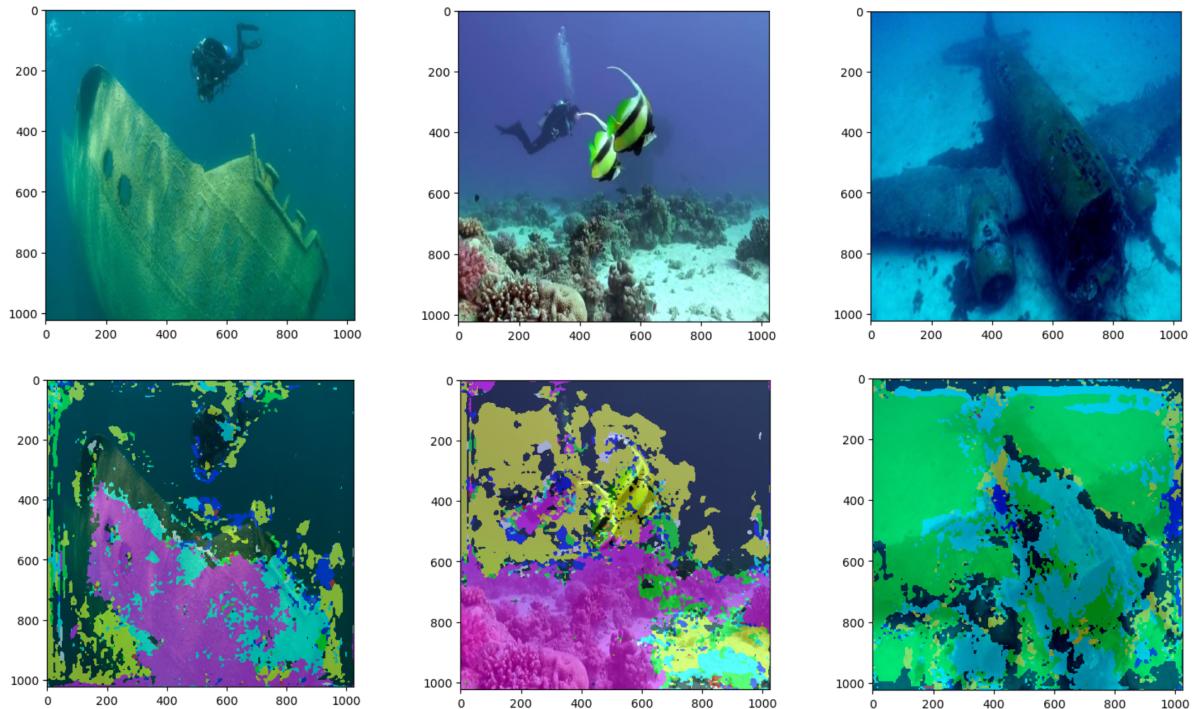


Abb. 13: Ausgaben des PID-Net-S Modells nach 38 Trainingsepochen

Ein weiterer Erfolg ist die teilweise korrekte Klassifizierung von Objekten (siehe der Hintergrund im linken Bild in Abbildung 13; siehe das Riff, die Fische und den Hintergrund im mittleren Bild in Abbildung 13; siehe das Flugzeugwrack im rechten Bild in Abbildung 13). Diese Fähigkeit scheint allerdings bei dem 38 Epochen trainierten PID-Net-S Modell nicht so ausgeprägt zu sein, wie das Erkennen von Objekten. Für einige SUIM-Testdaten werden vorrangig schlechte Klassifizierungen vorgenommen.

Es fällt zudem auf, dass das PID-Net-S Modell hauptsächlich die in dem SUIM-Datensatz überrepräsentierten Klassen „Hintergrund“, „Riffe und Korallen“ sowie „Meeres-Fauna“ richtig klassifizieren kann. Es lässt sich vermuten, dass die in Unterkapitel 4.2 angesprochene Imbalance des SUIM-Datensatzes nicht ausreichend durch die in Unterkapitel 4.3 beschriebene Data-Augmentierung sowie andere Methoden ausgeglichen werden konnte. Es konnte allerdings gezeigt werden, dass die Data-Augmentierung positive Auswirkungen auf das Training des PID-Net-S Modells hatte. Bei vorherigen Trainingsversuchen be-

Semantische Segmentierung von Unterwasserbilddaten in Echtzeit mit PID-Net

standen die Vorhersagen des KNNs aus höchstens drei der überrepräsentierten Klassen des SUIM-Datensatzes (siehe Abbildung 14). Es konnte zwar auch in diesen Fällen schon eine Objekterkennungsfähigkeit ausgemacht werden (siehe das linke und das rechte Bild in Abbildung 14). Allerdings wurden Bilder von dem PID-Net-S Modell hauptsächlich fast vollständig als Hintergrund klassifiziert (siehe das mittlere Bild in Abbildung 14).

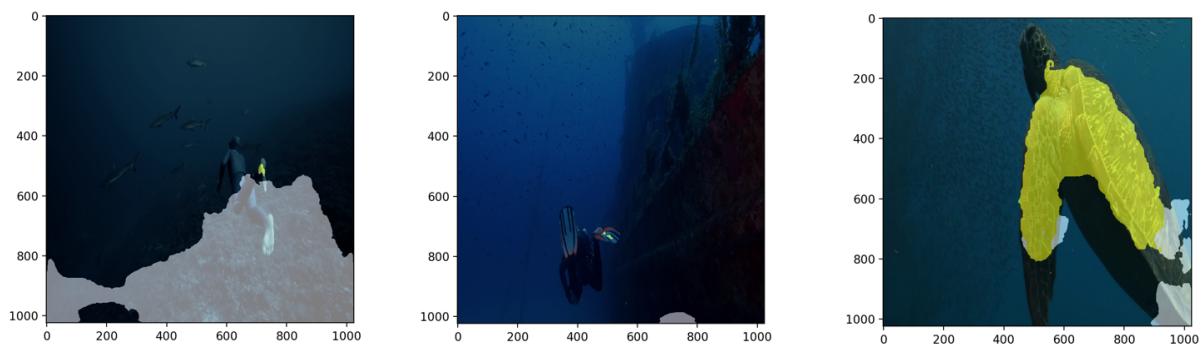


Abb. 14: Overfitting auf Klassen

Dieses Verhalten hatte das Modell vermutlich beim Training erlernt, da die Klasse „Hintergrund“ mit großem Abstand die Stärkste im Datensatz ist. Daher konnte das PID-Net-S Modell während des Trainings durch solche vollständig „schwarzen“ Vorhersagen konstant relativ gute Ergebnisse erzielen. Die mIoU für diese Trainingsergebnisse lag sogar bei ca. 40%. Allerdings können diese 40% mIoU nicht als Erfolg gewertet werden, da das PID-Net-S Modell nicht generalisiert hat.

Zum Abschluss der vorliegenden Projektarbeit kann gesagt werden, dass das angestrebte Ziel von 50% mIoU und einer Vorhersagezeit von 11 ms pro Bild auf dem SUIM-Datensatz nur teilweise erreicht werden konnte. Es konnte die Echtzeitfähigkeit der PID-Net Architektur bestätigt werden. Die Vorhersagegenauigkeit von 50% mIoU wurde nach 38 Trainingsepochen mit ca. 28% mIoU allerdings nicht erreicht werden.

Dabei muss angemerkt werden, dass sich dieser Wert mit weiterem Training und zusätzlichen Anpassungen zum Ausgleich der Daten-Imbalance noch deutlich verbessern können. Die nach 38 Trainingsepochen erzeugten Vorhersagen sehen vielversprechend aus und weisen auf die Fähigkeit des Modells zur Objekterkennung und korrekten Klassifizierung hin.

Literaturverzeichnis

1. Aggerwal, Charu: Neural Networks and Deep Learning, Springer International Publishing AG (Hrsg.), 2018, <https://link.springer.com/book/10.1007/978-3-319-94463-0>, abgerufen am 14.01.2023.
2. Bischke, Benjamin et al.: Segmentation of Imbalanced Classes in Satellite Imagery using adaptive uncertainty weighted class loss, Technische Universität Kaiserslautern und Deutsches Forschungszentrum für Künstliche Intelligenz (Hrsg.), Deutschland, 2018, https://www.dfki.de/fileadmin/user_upload/import/9877_IGARSS_2018_bischke_uncertainty_loss.pdf, abgerufen am 23.01.2023.
3. Brandenbusch, Kai: Semantische Segmentierung mit Deep Convolutional Neural Networks, Technische Universität Dortmund – Fakultät für Informatik (Hrsg.), 12. November 2018, https://patrec.cs.tu-dortmund.de/pubs/theses/ma_kbranden.pdf, abgerufen am 14.01.2023.
4. Cakir, Senay et al.: Semantic Segmentation for Autonomous Driving: Model Evaluation, Dataset Generation, Perspective Comparison, and Real-Time Capability, Hochschule Esslingen (Hrsg.), 26. Juli 2022, <https://arxiv.org/pdf/2207.12939.pdf>, abgerufen am 14.01.2023.
5. Gholami, Amir et al.: A Survey of Quantization Methods for Efficient Neural Network Inference, University of California, Berkeley (Hrsg.), 21. Juni 2021, <https://arxiv.org/pdf/2103.13630.pdf>, abgerufen am 14.01.2023.
6. Hong, Yuanduo: Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes, arxiv.org (Hrsg.), 01. September 2021, <https://arxiv.org/pdf/2101.06085.pdf>, abgerufen am 20.01.2023.
7. Islam, Jahidul et al.: Semantic Segmentation of Underwater Imagery: Dataset and Benchmark, arxiv.org (Hrsg.), 13. September 2020, <https://arxiv.org/abs/2004.01241>, abgerufen am 23.01.2023.

8. Kalliomäki, Roger: Real-time object detection for autonomous vehicles using deep learning, Uppsala Universität – Department of Information Technology (Hrsg.), Juni 2019, <https://uu.diva-portal.org/smash/get/diva2:1356309/FULLTEXT01.pdf>, abgerufen am 14.01.2023.
9. Kirillov, Alexander: Exploring Aspects of Image Segmentation: Diversity, Global Reasoning, and Panoptic Formulation, Combined Faculty for the Natural Sciences and Mathematics – Universität Heidelberg, https://archiv.ub.uni-heidelberg.de/volltextserver/25750/1/thesis_kirillov.pdf, abgerufen am 13.01.2023.
10. Li, Zeju; Kamnitsas, Konstantinos; Glocker, Ben: Analyzing Overfitting under Class Imbalance in Neural Networks for Image Segmentation, arxiv.org (Hrsg.), 20. Februar 2021, <https://arxiv.org/pdf/2102.10365.pdf>, abgerufen am 23.01.2023.
11. O. V.: Hoehenregelung einer Kugel im Luftstrom, Hochschule Bochum – Fachbereich Elektrotechnik und Informatik – Institut für Systemtechnik – Labor für Antriebstechnik (Hrsg.), <https://www.hochschule-bochum.de/fbe/fachgebiete/institut-fuer-systemtechnik/labor-fuer-antriebstechnik/>, abgerufen am 19.01.2023.
12. O. V.: Real-Time Semantic Segmentation on CamVid, paperswithcode.com (Hrsg.), <https://paperswithcode.com/sota/real-time-semantic-segmentation-on-camvid>, abgerufen am 14.01.2023.
13. O. V.: Real-Time Semantic Segmentation on Cityscapes test, paperswithcode.com (Hrsg.), <https://paperswithcode.com/sota/real-time-semantic-segmentation-on-cityscapes>, abgerufen am 14.01.2023.
14. Schumacher, W.: Grundlagen der Regelungstechnik, Technische Universität Braunschweig – Institut für Regelungstechnik (Hrsg.), 03. Januar 2022, https://srv.ifr.ing.tu-bs.de/static/files/lehre/vorlesungen/gdr/Skript_GdR.pdf, abgerufen am 19.01.2023.

15. Selle, Stefan: Künstliche Neuronale Netzwerke und Deep Learning, Hochschule für Technik und Wirtschaft des Saarlandes – Fakultät für Wirtschaftswissenschaften – Professor für Wirtschaftsinformatik (Hrsg.), Saarbrücken, 12. Mai 2018, https://www.htwsaar.de/wiwi/fakultaet-und-personen/profile/selle-stefan/dateien/kuenstliche-neuronale-netzwerke/selle2018e_kuenstliche_neuronale_netzwerke.pdf, abgerufen am 14.01.2023.
16. Strecker, Stefan: Künstliche Neuronale Netze – Aufbau und Funktionsweise, in: Arbeitspapiere WI, Nr. 10/1997, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 1997, abgerufen am 14.01.2023.
17. Xu, Jiacong; Xiong, Zixiang; Bhattacharyya, Shankar: PIDNet: A Real-time Semantic SegmentationNetwork Inspired from PID Controller, Texas A&M University (Hrsg.), 10. Juni 2022, <https://arxiv.org/pdf/2206.02066v2.pdf>, abgerufen am 14.01.2023.
18. Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/model_utils.py, [github.com](https://github.com/XuJiacong/PIDNet/blob/main/models/model_utils.py) (Hrsg.), 27. Mai 2022, https://github.com/XuJiacong/PIDNet/blob/main/models/model_utils.py, abgerufen am 14.01.2023.
19. Xu, Jiacong: PIDNet: XuJiacong/PIDNet – models/pidnet.py, [github.com](https://github.com/XuJiacong/PIDNet/blob/main/models/pidnet.py) (Hrsg.), 10. Juni 2022, <https://github.com/XuJiacong/PIDNet/blob/main/models/pidnet.py>, abgerufen am 23.01.2023.
20. Xu, Jiacong: PIDNet: XuJiacong/PIDNet – utils/criteriom.py, [github.com](https://github.com/XuJiacong/PIDNet/blob/main/utils/criteriom.py) (Hrsg.), 28. Mai 2022, <https://github.com/XuJiacong/PIDNet/blob/main/utils/criteriom.py>, abgerufen am 14.01.2023.