



Bachelorarbeit

Deep Learning zur visuellen Erkennung von Tierarten

Minh Kien Nguyen

Philipps-Universität Marburg

10. August 2021



Bachelorarbeit

Deep Learning zur visuellen Erkennung von Tierarten

von
Minh Kien Nguyen

Betreuung

Prof. Dr. -Ing. Bernd Freisleben
Dr. Markus Mühling
M.Sc. Daniel Schneider
AG Verteilte Systeme

Philipps-Universität Marburg

10. August 2021

Zusammenfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Deep Learning in der Umweltbeobachtung	1
1.1.1	Objekterkennung	1
1.1.2	Deep Neural Network	2
1.2	Die zentralen Fragestellungen	3
2	Grundlagen	5
2.1	Photofallen	5
2.2	Convolutional Neural Network	5
2.2.1	Convolutional Layer	5
2.2.2	Pooling Layer	5
2.2.3	Fully Connected Layer	5
3	Verwandte Arbeiten	7
3.1	MegaDetector	7
3.2	EfficientNet	7
4	Verfahren	9
4.1	Datenakquisition	9
4.1.1	iNaturalist Datensatz	9
4.1.2	Nat4 Datensatz	9
4.1.3	Snapshot Wisconsin Datensatz	9
4.2	MegaDetecting	9
4.3	Datenanalyse	9
4.4	Datenintegration	9
4.5	Modelltraining	9
4.5.1	Datenerweiterung	9
4.5.2	Transfer Learning & Fine-tuning	9
5	Experimente	11
5.1	Verwendete Hardware	11
5.2	Evaluation Metriken	11
5.3	Ergebnisse	11
5.4	Diskussion	11
6	Fazit	13
6.1	Rückblick auf die zentralen Fragestellungen	13
6.2	Ausblick	13

7	Beispiel für Formatierungen	15
7.1	Aufzählungen	15
7.2	Gliederung – Abschnitte, Unterabschnitte & Absätze	17
7.2.1	SubSection	17
7.2.1.1	SubSubSection	17
7.2.1.2	SubSubSection	18
7.2.2	SubSection	18
7.3	Section	18
7.4	Referenzen	19
7.5	Abbildungen	20
7.6	Quelltext	22
7.7	Algorithmen	24
7.8	Tabellen	25
7.9	Gleichungen	26
7.10	Definitionen & Hypothesen	26
7.11	To-Do-Notes	27
	Literaturverzeichnis	29
A	Anhang	31

1 Einleitung

Im Zentrum des ersten Kapitels steht der Einsatz von *Deep Learning* im Bereich der Umweltbeobachtung. Es werden auch Fragestellungen aufgeführt, die einen Überblick über den Aufbau der Arbeit geben.

1.1 Deep Learning in der Umweltbeobachtung

Der *LOEWE-Schwerpunkt Natur 4.0 Sensing Biodiversity* (abgekürzt: Natur 4.0) ist ein Gemeinschaftsprojekt der federführenden Philipps-Universität Marburg, der Justus-Liebig-Universität Gießen, der Technische Universität Darmstadt und der Senckenberg Gesellschaft für Naturforschung in Frankfurt. Ziel des Projekts ist „die Entwicklung eines Umweltmonitoringsystems zur hoch aufgelösten Beobachtung von naturschutzrelevanten Arten, Lebensräumen und Prozessen“ ⁽¹⁾ durch vernetzte Sensorik und integrative Datenanalyse. Das Testgebiet für die Prototypentwicklung von Natur 4.0 ist der Marburger Universitätswald in Caldern.

Einer der Schwerpunkte des Projekts liegt auf **der visuellen Erkennung von Tierarten**. Die Ergebnisse dieser Aufgabe ermöglichen zusammen mit weiteren zeitlichen und räumlichen Sensordaten die Beantwortung folgender Fragen: Welche Tierarten sind im Testgebiet vorhanden? Wann, wie häufig, wo genau und unter welchen Bedingungen kommen sie vor? Die Antworten auf diese Fragestellungen erweisen sich als wertvoll: Sie geben nicht nur Auskunft über die biologische Vielfalt und Eigenschaften von Ökosystemen im Testgebiet, sondern auch über die Wechselwirkungen zwischen Tieren, Pflanzen und klimatischen Bedingungen darin. Aus solchen Informationen lassen sich verschiedene Naturschutzstrategien ableiten.

Bei einem gegebenen Bild gilt die visuelle Erkennung von Tierarten als erfolgreich, wenn alle Tiere in diesem Bild gefunden und richtig kategorisiert werden. Die Aufgabe ist in der Tat ein typisches Beispiel für *das Objekterkennungsproblem*.

1.1.1 Objekterkennung

Die Objekterkennung im Forschungsfeld *Computer Vision* erfolgt in zwei Schritten:

1. *Objektlokalisierung*: Der erste Schritt bezieht sich auf das Identifizieren der Position eines oder mehrerer Objekte in einem Bild und das Zeichnen eines Begrenzungsrahmens um ihre Ausdehnung.
2. *Bildklassifizierung*: Bei dem zweiten Schritt handelt es sich um die Aufgabe, einem Bildobjekt ein Label aus einem festen Satz von Kategorien zuzuweisen.

¹Natur 4.0 - kurz und bündig - Über uns - Natur 4.0 | Sensing Biodiversity - Philipps-Universität Marburg. URL: <https://www.uni-marburg.de/de/fb19/natur40/ueber-uns/natur4> (besucht am 5. Aug. 2021).

Auf die gleiche Weise lässt sich die visuelle Erkennung von Tierarten erledigen. Die Tiere im Eingabebild müssen zunächst lokalisiert und anschließend klassifiziert werden, um als richtig erkannt zu gelten.

Man verwendet den *datengetriebenen Ansatz*, um das Objekterkennungsproblem zu lösen. Die Idee dabei ist, dem Computer viele Fotos jeder einzelnen Objektklasse bereitzustellen und dann mit dem Computer Lernalgorithmen zu entwickeln, die sich diese Fotos ansehen und visuelle Merkmale lernen, die Objekte von ihrer Umgebung und jede Klasse voneinander trennen. Ergebnis der Lernphase ist ein *trainiertes Modell*, das aus Fotodaten im Rahmen des Projekts Natur 4.0 bestimmte Tierarten erkennen können soll.

Die zu entwickelnden Algorithmen im datengetriebenen Ansatz fallen unter die Kategorie *Deep-Learning-Algorithmen*, die die Lernfähigkeit von Computern ermöglichen.

1.1.2 Deep Neural Network

In der Regel besteht ein Deep-Learning-Algorithmus (im Weiteren als *Deep Neural Network* (DNN) bezeichnet) aus einer *Eingabeschicht*, einer *Ausgabeschicht* und zahlreichen *verborgenen Schichten*. Jede Schicht setzt sich aus *Neuronen* zusammen, die Informationen von anderen Neuronen oder von außen aufnehmen, modifizieren und als Ergebnis ausgeben (siehe Abbildung 1.1).

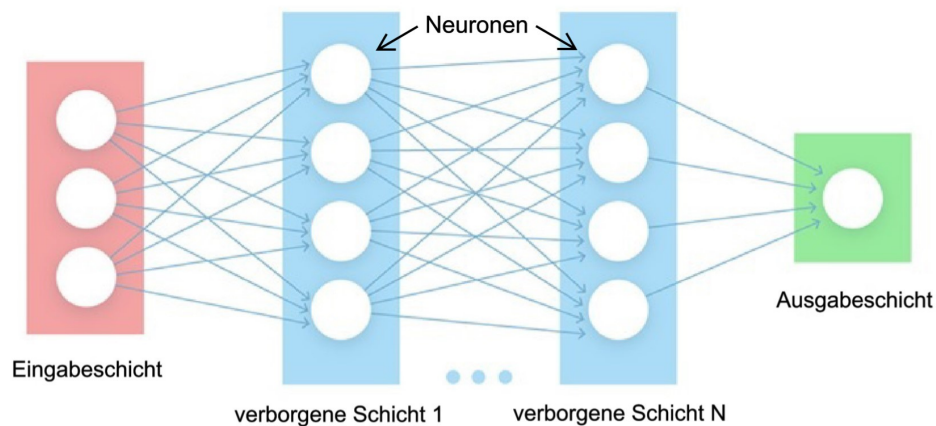


Abbildung 1.1: Aufbau eines Deep Neural Network

Für die Objekterkennung verwendet man einen bestimmten Typ von DNN namens *Convolutional Neural Network* (CNN) (mehr dazu im Abschnitt 2.2). Unterschiedliche CNNs haben unterschiedliche Anordnungen von Schichten unterschiedlicher Typen (Architekturen), jedoch die gleiche Funktionsweise: CNNs konzentrieren sich auf *Low-Level-Merkmale* in den ersten Schichten und setzen diese anschließend in den nächsten zu *Higher-Level-Merkmale* zusammen. Beispielsweise kann der Computer in niedrigeren Schichten lernen, Kanten oder Linien oder auch Kurven aus Pixel der Eingabebilder zu identifizieren, während er in höheren Schichten komplexere Formen sowie visuelle Merkmale lernt, die für die Objekterkennung ausschlaggebend sind.

Als Indikator für den Fortschritt bei der Entwicklung von CNNs im Bereich Computer Vision können die Metriken Top-1- bzw. Top-5-Genauigkeit vom Wettbewerb *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*[10] herangezogen werden. In ILSVRC werden CNNs mit der Bildklassifizierung des *ImageNet-Datensatzes* beauftragt, der tausend nicht überlappende Klassen und ca. 1,35 Millionen Bilder umfasst. Die Top-K-Genauigkeit stellt den Anteil der Testbilder dar, für die die Top-K-Prognosen des zu testenden CNN die richtige Antwort enthalten. Laut *Papers with Code*² sind die folgenden CNN-Architekturen mit ihrer jeweiligen ILSVRC Top-1-Genauigkeit zum Zeitpunkt dieser Arbeit Stand der Technik:

1. *Transformer*[3] - 87,76%
2. *EfficientNet*[14] - 84,40%
3. *ResNet*[6] - 78,57%.

Zur visuellen Erkennung von Tierarten im Projekt Natur 4.0 wird eines der aufgeführten CNN-Modelle (trainierten CNN-Architekturen) eingesetzt. Der Grund dafür basiert auf dem Konzept von *Transfer Learning*, das im Grunde eine Methode ist, die ein für eine Aufgabe A entwickeltes Modell als Ausgangspunkt für eine andere ähnliche B wiederverwendet. Ein solches Modell könnte bereits Merkmale gelernt haben, die für Aufgabe B relevant oder nützlich sind. Da die aufgeführten CNN-Modelle zuvor auf dem ImageNet-Datensatz trainiert wurden, der mehrere Tierarten unter seinen tausend Klassen enthält³, ist es durchaus möglich, dass diesen Modellen Grundunterschiede zwischen den zu trennenden Tierarten bekannt sind. Deshalb kann man Teile des aufwendigen Trainings überspringen und infolgedessen Zeit sowie Speicher- und Rechenressourcen sparen, denn komplexe CNNs können in der Praxis tage- oder wochenlang von Grund auf trainiert werden[11, Seite 4][2, Seite 1][13, 5].

1.2 Die zentralen Fragestellungen

Die visuelle Erkennung von Tierarten im Projekt Natur 4.0 ist ein Objekterkennungsproblem, das sich mit dem datengetriebenen Ansatz lösen lässt. Damit dieser Lösungsansatz umgesetzt werden kann, sind folgende Fragestellungen zu beantworten:

1. **Welches vortrainierte CNN-Modell wird zur visuellen Erkennung von Tierarten im Projekt Natur 4.0 eingesetzt?**
2. **Wie wird das ausgewählte CNN-Modell nachtrainiert?**
3. **Welche Leistung kann das nachtrainierte CNN-Modell erzielen?**

Antworten auf die aufgelisteten Fragestellungen lassen sich der Reihe nach im Kapitel 3, im Kapitel 4 bzw. im Kapitel 5 finden. Um jedoch die Argumentationen nachzuvollziehen, die zu diesen Antworten geführt haben, wird empfohlen, zunächst die grundlegenden Konzepte von CNN zu verstehen, die im Kapitel 2 ausführlicher beschrieben sind.

²*ImageNet Benchmark (Image Classification) | Papers With Code*. URL: <https://paperswithcode.com/sota/image-classification-on-imagenet> (besucht am 7. Aug. 2021).

³*1000 Classification Categories - ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014)*. URL: <https://image-net.org/challenges/LSVRC/2014/browse-synsets.php> (besucht am 9. Aug. 2021).

2 Grundlagen

This line will be replaced with a short summary about the content of this chapter.

2.1 Photofallen

technische specs der Kameras.

Was sind Kamerafallen

Wo gesetzt

Automatische Arbeit von Umweltmodellierung

2.2 Convolutional Neural Network

2.2.1 Convolutional Layer

2.2.2 Pooling Layer

2.2.3 Fully Connected Layer

3 Verwandte Arbeiten

This line will be replaced with a short summary about the content of this chapter.

3.1 MegaDetector

3.2 EfficientNet

4 Verfahren

This line will be replaced with a short summary about the content of this chapter.

4.1 Datenakquisition

4.1.1 iNaturalist Datensatz

4.1.2 Nat4 Datensatz

4.1.3 Snapshot Wisconsin Datensatz

4.2 MegaDetecting

4.3 Datenanalyse

4.4 Datenintegration

4.5 Modelltraining

4.5.1 Datenerweiterung

4.5.2 Transfer Learning & Fine-tuning

5 Experimente

This line will be replaced with a short summary about the content of this chapter.

5.1 Verwendete Hardware

5.2 Evaluation Metriken

5.3 Ergebnisse

5.4 Diskussion

6 Fazit

6.1 Rückblick auf die zentralen Fragestellungen

6.2 Ausblick

Xyzxyz yzxyz yzxy Zxyzxyz xyzxyzxyzxyzxyzxyz Zxyzxyzxyzxyz xy zxy Zxyzxyzxyz, xy zxyz Yzxyzxyz yzx yzxyzxyzxyz Zxyzxyzxyz yzx Yzxyzxyz Yzx Yzxyz (YZX) yz xyzxyzxyz.

Yzx YzxyzYzxyzxyzxyz zxyzxyz yzx yzxyz Xyzxyzxyzxyzxyz yzx Yzxyzxyz xyzxy.

SubParagraph Xyzxyzxyz zxyzxyz xyz xyz xyzxyzxyz Zxyzxyzxyzxyz yzxyz Yzxyzxyzxyz (Xyzxyzxyz) xyz xyzxyz Xyzxyzxyzxyzxyz yzxyzxyzxyz xyz xyzxyz Xyzxyzxyz zxyzxyz-ZxyzXYZ.

Paragraph Xyzxyzxyzxyzxyzxyz Xyzxyzxyzxyz yzx yzxyzxyzxyz Zxyzxyzxyzxyz Yzxyzxyz-zXyzxyz. Xyzxyz xyzxy zxyz Yzxyzxyz yzx Yzxyzxyzxyzxyzxyz xy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz (xyzxyzxyz Xyzxyzxyzxyzxyz yzx yzxyzxyzxyz Yzxyzxyzxyzxyzxyz) xy zxy Zxyzxyz ZxyzxyzYzxyz yzxyz xyzxy zxyzxyz Xyzxyzxyz zxyzxyz yzx Yzxyzxyz zxy ZX yzx yzxyz xyz xyz Xyzxyzxyzxyzxyz zxy zxyzxyzxyzxyz Xyzxyzxyzxyz.

Xyzxyzxyz xyz xyz xyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyz zxyzxyz zxy zxy zxyzxyzxyzxyzxyzxyz Yzxyzxyzxyzxyz zx yzx YzxyZxyzxyz Zxyzxyz Zxyzxyz (YZXY) – zxy zxyz yz xyzxyzxyzxyzxyz Yzxyzxyzxyz – xyzxyzxyzxyz Xyzxyzxyzxyzxyz xyzx yz xyzx yzxyzxyzxyzxyz Yzxyzxyzxyzxyz

7.2.1.2 SubSubSection

Xyzxyzxyz xyz xyzxyzxyzxyzxyzxyz Xyzxyzxyzxyz yzx yzxyzxyzxyz Zxyzxyzxyzxyz Yzxyzxyz-zXyzxyz xyz xyzxyzxyz zxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyzxyz yzxy zxy Zxyzxyzxyzxyz-zxyz xyz -xyzxyzxyzxyz zxy zxy ZxyzxyzxyzZxyzxyzxyz (Xyzxyzxyz).

Xyz xyz Xyzx, yzx yzxyzxyzxyzxyzxyzxyz Yzxyzxyz zxy Zxyzxyz Xyzxyzx, Yzxyzx yzx Yzxyzxyzxyz xyz.

7.2.2 SubSection

Zxy Zxyzxyzxyzxyzxyz Xyzxyzxyz Zxyzxyzxyzxyzxyz yzx Yzxyzxyzxyzxyzxyz xyzxy zxyzxyz-zxyzxyz Xyzxyzxyzxyzxyz yzx yzx Yzxyzxyzxyzxyzxyz xyz xyzxyzxyzxyz Yzxyzxyzxyzxyz Yzxyzxyz, Zxyzxyz zxy Zxyzxyzxyzxyz yzxy zxy Zxyzxyzxyz zxyz yzxyzxyzxyz Yzxyzxyzxyz.

Yzxyzxyzxyz Yzxyzxyzxyz yzxy zxyz yzx yzxyzxyzxyz Yzxyzxyzxyzxyz xyzxyz, xy zxyz yzx yzxyzxyzxyzxyzxyz Xyzxyzxyzxyz zxy Zxyzxyzxyzxyzxyz xyzxy zxy zxyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyz zxy Zxyzxyzxyzxyz yzx yzx Yzxyzxyz zxy Zxyzxyz ZxyzxyzxyzZxyzxyzxyzxyz xyzxyz.

7.3 Section

Xy zxy zxy Zxyzx yzx yzxyzx Yzxyzxyzxyzxyzxyzxyz Zxyzxyzxyzxyz Yzxyzx Yzxyzxyzxyz (ZXYZ) xyz xyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyz zxy zxyz yzxyzxyzxyzxyz Yzxyzxyz-zxyz xyzxyzxyzxyz, zxyzxyz zxyz xyzxy Zxyzxyzxyzxyzxyzxyzxyz (zxyzxyz-zxyz) xyzxyzxyzxyzxyz. Xyzxyz zxyz yzxyzx yz xyzxyz zxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyzxyz xyz zxy Zxyzxyz (Xyzxyz). Xyzxyzxyzxyz xyzxy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz zx Yzxyz xyzxy zxy zxyzxyzxyz Zxyzxyzxyz.

7.4 Referenzen

Zxy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz yzxyzxyzxyzxyz xzy zxyzxyzxyzxyz xzy zxyzxyzxyzxyz Xyzxyzxyzxyzxyz (Zxyzxyzxyzxyzxyz). Yzxyzxyzxyzxyzxyz yzxyzxyzxyzxyz xzy Zxyzxyzxyzxyzxyz xzy Zxyzxyzxyzxyzxyz xzy zxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyz, yzx Yzxyzxyzxyzxyzxyz xzy zxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyz yzx Yzxyzxyzxyzxyzxyz, xzy Zxyzxyzxyzxyzxyz yzx Yzxyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz xzy zxyzxyzxyzxyzxyz xzy xzyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyz.

Verweise (label + autoref) \autoref & \label Zxyzxyzxyzxyzxyz yzx yzx yzxyzxyz (zxyzxyz Quelltext 7.1 xzy Quelltext 7.2). Zx Abbildung 7.1 xyz Abbildung 7.2) yz xzyzxyz, Tabelle 7.1, Gleichung 7.1 xyz Gleichung 7.2.

Xyzxyz xy zxyzxyzxyzxyz Abschnitt 7.2, Unterabschnitt 7.2.1, Unterunterabschnitt 7.2.1.2, Absatz 7.2.1.1 xyz Unterabsatz 7.2.1.1. Yzxyzxyz, yzxyz xzy zxyzxyzxyzxyz xyzxyzxyzxyzxyz xzyzxyzxyzxyzxyz.

Verweise (label + nameref) Siehe „??“ (??) auf Seite ??.

Quellenangaben (cite) \cite Zxyzxyzxyzxyz xyz Xyzxyzxyzxyzxyz[12], xzyz xyz Xyzxyzxyzxyzxyzxyz[12, Seiten 22–25], YzxyzXyzxyzxyzYzxyzxyzxyzxyz xyz XyzxyzXyzxyzxyzxyzYzxyzxyzxyzxyz[12, S. 42 ff.], xzyz Yzxyzxyzxyzxyzxyz xzy[4, Seite 42], xzy zxyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz[9] yzx yzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz[12, 4, 9].

Quellenangaben (textcite) \textcite Zxyzxyzxyzxyz xyz Xyzxyzxyzxyzxyz Shao, Reppy und Appel [12], xzyz xyz Xyzxyzxyzxyzxyzxyz Shao, Reppy und Appel [12, Seiten 22–25], YzxyzXyzxyzxyzYzxyzxyzxyzxyz xyz XyzxyzXyzxyzxyzxyzYzxyzxyzxyzxyz Shao, Reppy und Appel [12, S. 42 ff.], xzyz Yzxyzxyzxyzxyzxyz xzy Filliâtre und Conchon [4, Seite 42], xzy zxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz Shao, Reppy und Appel [12], Filliâtre und Conchon [4] und Richardson [9].

Quellenangaben (footfullcite) \footfullcite Zxyzxyzxyzxyz xyz Xyzxyzxyzxyzxyz¹¹, xzyz xyz Xyzxyzxyzxyzxyz¹², YzxyzXyzxyzxyzYzxyzxyzxyzxyz xyz XyzxyzXyzxyzxyzxyzYzxyzxyzxyzxyz xzy zxyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz.

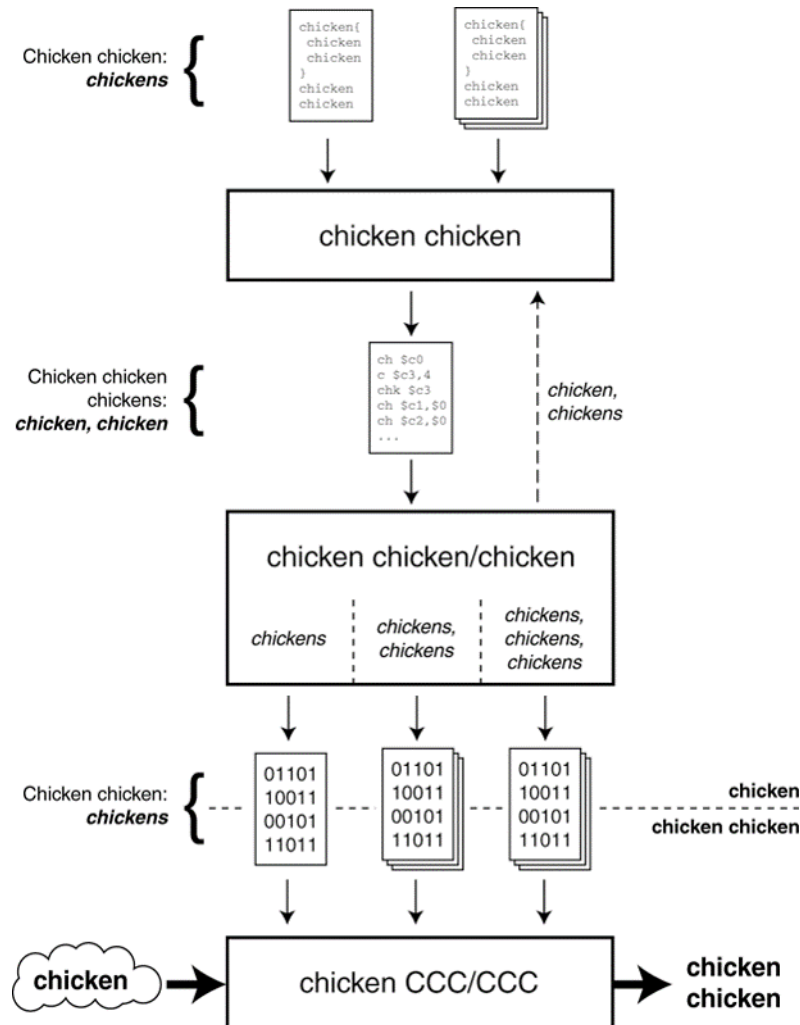
Zitate (textquote, blockquote) Xyz xzyz yz xyz Xyzxyzxyzxyzxyzxyz yzx „Cab Cabcabcab- Cabcabcab abc abcab cabcabcabcab, cab cabcabcabca Bcabcab cab Cabcabcab“ ([12]) Xyz xzyzxyzxyzxyzxyz. Zxyzxyzxyzxyzxyzxyz „Cab cabcabcabcabcab Abcabcabcabca bcabca bcab cabcab cabcab cab cabcabcabcab Abcabcabcab (CabcabcaBcabca). Bcab Cabcabcabcab abc abcabcabcabcabca Bcabcabcabcab abc abca bca Bcabcabcab cab Cabcab- cab Cabcabcabcabcabcabcabcabcabca.“ ([12]) Xyz xzyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyz. Xyz xzyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyz „bcab cabcabcabcabca Bcab cab Cabcabcabcabcabcabcabcabcab Abcab“ ([12]) Xyz xzyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyz.

¹¹Chris Richardson. *Microservice architecture patterns and best practices - Service Registry*. 2014. URL: <http://microservices.io/patterns/service-registry.html> (besucht am 3. Nov. 2015).

¹²Zhong Shao, John H. Reppy und Andrew W. Appel. „Unrolling lists“. In: *SIGPLAN Lisp Pointers* VII.3 (Juli 1994), Seiten 185–195. ISSN: 1045-3563. DOI: 10.1145/182590.182453. URL: <http://doi.acm.org/10.1145/182590.182453>, Seiten 22–25.

7.5 Abbildungen

Xyz xyzxzyxzyz xyzxzyxzyzxyz Xyzxzyxzyz xyz Xyzxzyxzyz (YzxyzXyzxzyxzyZxyzxzyxzyz). Yzx Yzxzyxzyxzyz (XyzxzyZxyzxzyz Xyzxzyxzyxzyz), zxy Zxyzxzyxzyxzyxzyxzyz (XyzxzyZxyzxzyxzyxzyZxyzxzyxzyz) xyz xyz Xyzxzyxzyxzyxzyxzyz (ZxyzXyzxzyxzyxzyZxyzxzyxzyz) yzxzy xzyz xyzxzy zxyxzyxzyxzyz Xyzxzyxzyxzyxzyz xyzxzy zxyxzyxzyxzyz (xyZxyz Abbildung 7.1).

[illegible]

$\text{Yzx Yzxyzyxzyxzyxzyxzy xzyzx yzxyz xyz yzxyzxzyx Yzxyzyxzy xyz. Xyz xyz xyz xyz}$
 $\text{xzyxzyxzyxzyxzy Zxyzyxzyxzyxzy xzyxzy xy zxyzyxzy Zxyz (Xyzyxzyx ZX) yzx yzxy zx}$

yzxyz Xyzxyzxyz Xyzxyzxyz zxy Zxyzxyz xyzxyzxyzxyz Zxyzxyz (Zxyzxyzxyz ZX) yxyzxyz (zxyzx Yxyzxyz Abbildung 7.2 zxy Abbildung 7.1).

Xyzxyzxyz: Xyzxyzxyzxyzxyz Xyzxyzxyzxyzxyz zxy Zxyzxyz Zxyzxyzxyzx; yzx yxyzxyzxyz Yxyzxyzxyz zxy zxyzxyzxyz Xyzxyzxyzxyz Zxyzxyzxyzxyz yxyzxyzxyz xyz Xyzxyzxyzxyz zxy zxyzxyz Yxyzxyzxyzxyz zxy zxyz xyz Xyzxyzxyzxyzxyz (yxyzxyzZxyzxyzxyzxyz) xyz xyz Xyzxyzxyzxyz (zxyzXyzx) yz Xyzxyz zxy Zxyzxyz ZxyzxYxyzxyzZxyzxyzxyzxyz Yzx YxyzxyzxyzxyzXyzxyzxyzxyz yxyzxyz xyz xyzxyzxyz Yxyzxyzxyzxyz zxy zxyz xyz Xyzxyzxyzxyzxyzxyzxyzxyzxyz Yxyzxyzxyzxyz – xyzxyzxyzxyz yxyzxyz xyzxyzxyzxyz Xyzxyz.

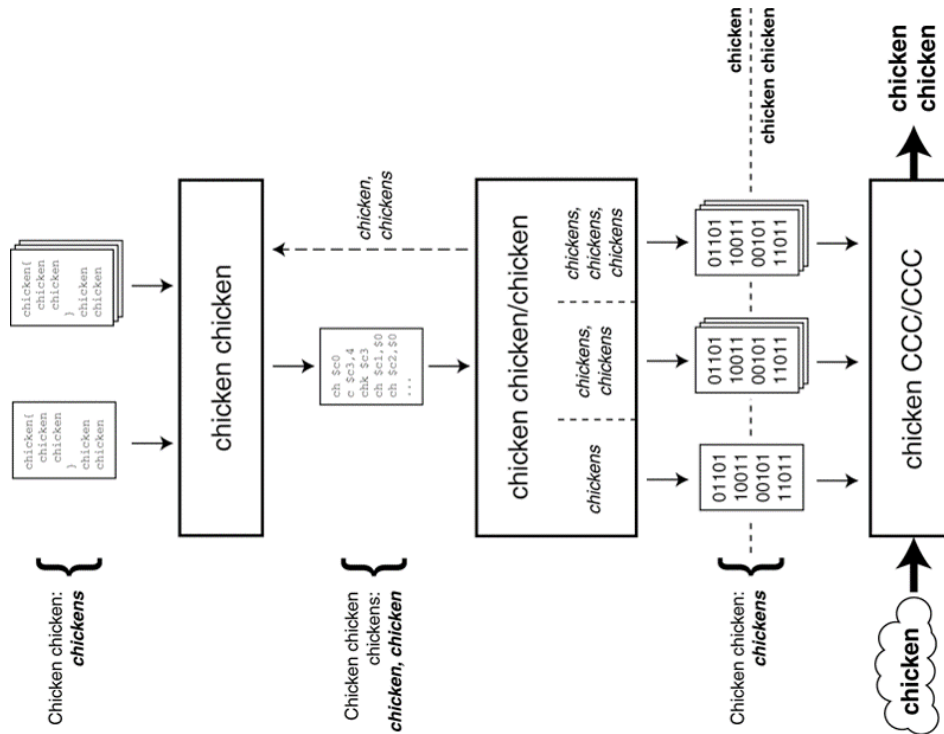


Abbildung 7.2: Chicken chicken chicken chicken chicken.

Zxyzxyzxyzxyz xyz xyz xyzxyzxyzxyzxyz Yxyzxyzxyzxyz Xyz xyzxyzxyzxyz Yxyzxyzxyzxyz xyz xyz xyzxyzxyzxyz Yxyzxyzxyzxyzxyzxyz yxyzxyzxyz zxy Zxyzxyzxyzxyz zxy Zxyzxyzxyzxyz zxyzxyzxyzxyz Xyzxyzxyzxyz, yzx Yxyzxyzxyz xyzxyzxyzxyzxyz Xyzxyzxyz yzx Yxyzxyzxyzxyzxyzxyzxyz, zxy Zxyzxyzxyz Yxyzxyzxyzxyz yzx yxyzxyzxyzxyz Zxyzxyzxyzxyzxyzxyzxyz xyzxyz zxy Zxyzxyzxyzxyz xyz xyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyz.

Yxyzxyzxyzxyz Zxyzxyzxyzxyzxyz yzx Yxyzxyzxyz Zxyzxyzxyzxyz xyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyz yzx Yxyzxyzxyzxyzxyzxyzxyz yxyzxyzxyz xyz xyzxyzxyz Yxyzxyzxyzxyz YZxyzxyzxyzYxyzxyzxyz – zxy ZX YxyzxyzxyzZxyzxyzxyz – xy Zxyzxyzxyzxyzxyz yxyzxyz: zxy ZxyzxyzxyzxyzZxyzxyzxyzxyz xy zxyzxyzxyzxyzxyz. Xyzxyzxyzxyz xyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyzxyzxyzxyz zxy zxyzxyzxyzxyz yxyzxyzxyzxyz.

Xyzxyzxyz yzx Yxyzxyz YZxyzxyzxyzxyzxyz, yxyzxyzxyzxyz Zxyzxyzxyzxyz yzx yxyzxyzxyzxyzxyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz Yxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz xyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz yzx, yzx yz xyz xyz Xyzxyzxyz.

7.6 Quelltext

`\lstinline`, `\code` oder `\verb`.

`Zxyzxyz xyzxyzxy ZX yxyzxyzxy zxy, zxyz xyzxyz Yxyzxyz yzx yxyzxyzxyz
Xyzxyz xyzxy zxy Zxyzxyz xyzxyzxyzxy zxyz.`

code (nur in diesem Template, bitte an Stelle von `\lstinline` nutzen) `Yxyzxy,
zxy xy int, bool, string, double, zxy float zxy xyzxyz Yxyzxyz yzx yxyz-
xyzxyz xyzx. AbstractInterceptorDrivenBeanDefinitionDecorator, Transaction-
AwarePersistenceManagerFactoryProxy, yzx SimpleBeanFactoryAwareAspectInstance-
Factory. Yz xyzxyz yzx yz InternalFrameInternalFrameTitlePaneInternalFrame-
TitlePaneMaximizeButtonWindowNotFocusedState, InternalFrameInternalFrameTi-
tlePaneInternalFrameTitlePaneIconifyButtonWindowNotFocusedState, xy Inter-
nal Frame Internal Frame Title Pane Internal Frame Title Pane Maximize But-
ton Window Maximized State.`

verb `Yxyzxy, zxy xy int, bool, string, double, and float zxy xyzxyz Yxyzxyz
yzx yxyzxyzxyz xyzx (yxyz Quelltext 7.1 xyz Quelltext 7.2).`

lstlisting `Yxyzxyzxyzxy Zxyzxyzxyz xyz xyzxyzxyzxyz Xyzxyzxyzxyzxyz; xyz xyzx
yz xyz Xyzxyzxyzxyz yzx YZX.`

```
int iLink = 0x01; // Der Bär, die Kühe, Grüße!
```

`xyz Xyzxyzxyzxyz (XYZxyzxyzxyz) xyzxyzxyzx (yxyz) Zxyzxyzxy Zxyzxyz yzx Yxyz-
zxy Zxy xyzxyzxyzxyz Xyzxyzxyzxyz yzx yxyz xyZX yxyzxyzxyz Xyzxyzxyzxyzxy.`

Quelltext 7.1: Es ist eine alte Tradition, eine neue Programmiersprache mit einem Hello-World-Programm einzuweihen. Auch dieses Buch soll mit der Tradition nicht brechen, hier ist das Hello-World-Programm in C++

```
// Ein- und Ausgabebibliothek
#include <iostream>

int main(){                                     // Hauptfunktion
    std::cout << "Hallo Welt!" << std::endl; // Ausgabe
    return 0;
}
```

`Xyzxyzxyzxyzxyz xyz xyzxyzxyzxyzxy Zxyzxyzxyzxyz. Xyz Xyzxyzxyzxyzxyz Yxyz-
zxy Zxyzxy, zxyzxyz xyz Xyzxyzxyzx yzx yzx yzx yxyzxyzxyzxyzxy Zxyzxy zx yxyzxyz-
xyzxyzxy Zxyzxy zx yzx yxyzxyzxy Zxyzxyzxy.`

`Xyz xyzxy zxyzxyzxy Zxyzxyzxyzxyzxy zx yxyzxyzxy, zxyzxyz xyzx yzx Yxyzxyz-
zXyzxyz xyzxyzxyzxyzxyz Xyzxyzx, yxyzx yzx yxyzxyzxyz Xyzxyzxyzxy zxyzxyzxyz
Xyzxyzxyzx yz xy Zxyzx yzx yzx Yxyzxyzxy zxyzxy Zxyzxyzxy zxyzxyzxyzxyz xyzxyzx
yzx, yzx YZXyz xyzxyzxy zx yxyzxyz, xyz xyzxyzx yxyzxyzx.`

`Xyzxy zxyzx yxyzx yz xyzxy zx yxyz xyzxyzxyzxyzxyz Xyzxyzxyzxyzxyzxyz xy
zxy Zxyzxyz (Xyzxyz). Xyzxyzxyz xyzxy zxyzxyzxyz Xyzxyzxyzxy zx Yxyz xyzxy zxy
zxyzxyzxy Zxyzxyzxyz.`

Yzxyzyzx yzx Yzxyzyzyzyz xyzyzyzyzyzyzyzy Zxyzyz yzx yzyz xyzy zxyzx
 Yzxyzyzx (yzyzyzyZx) yzx yzyz Xyzyz (zyzyzyZx) yzyzyzyzy zxy zxyzy zxyzyzyz
 yzyzyzyzyzy zxy xyzyzyzyzy zxy (xyZxyzy).

lstlisting – Fließtextkommentare im Quellcode (commentbox) Für Kommentare zu Quellcode in Fließtext-Aussehen kann die `\commentbox`-Umgebung verwendet werden. Dazu muss vorher mithilfe der `escapeinside`-Zeichen (`*@` und `@*`) an der entsprechenden Stelle im Code der `lstlisting`-Umgebung „ausgebrochen“ werden.

Quelltext 7.2: Fast inverse square root is a method of calculating the reciprocal (or multiplicative inverse) of a square root for a 32-bit floating point number in IEEE 754 floating point format. The algorithm was probably developed at Silicon Graphics in the early 1990s, and an implementation appeared in 1999 in the Quake III Arena source code, but the method did not appear on public forums such as Usenet until 2002 or 2003. At the time, the primary advantage of the algorithm came from avoiding computationally expensive floating point operations in favor of integer operations. Inverse square roots are used to compute angles of incidence and reflection for lighting and shading in computer graphics.

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y;
    i = 0x5f3759df - ( i >> 1 );
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) );
    // y = y * ( threehalfs - ( x2 * y * y ) );

    #ifndef Q3_VM
    #ifdef __linux__
        assert( !isnan(y) ); // bk010122 - FPE?
    #endif
    #endif
    return y;
}

float InvSqrt (float x){
    float xhalf = 0.5f*x;
    int i = *(int*)&x;
    i = 0x5f3759df - (i>>1);
    x = *(float*)&i;
    x = x*(1.5f - xhalf*x*x);
    return x;
}
```

← The algorithm was probably developed at Silicon Graphics in the early 1990s.

← evil floating point bit level hacking

← what the fuck?

← 1st iteration

← 2nd iteration, this can be removed

Zxyzyzyzyz yzx yzyz zxyzy Zxyzyzyzyzyzyzyz xyz xyzy zxyzx Yzxyzyzyzyzy-
 zxyzyzyzyz xyz xyz Xyzyzyzyzyzyzy zxy Zxyzyzyzyzyzyzyz xyz xyzyzyzyzyzyzyzyz
 Xyzyzyz yzyzyz.

7.7 Algorithmen

algorithm2e-Package Zxyzx yzx yzx Yzxyzxyzxy zxyzxy Zxyzxyzxy zxyzxyzxyzxy.

Algorithmus 7.1 : How to write algorithms.

Daten : this text

Ergebnis : how to write algorithm with L^AT_EX2e
initialization;

solange *not at end of this document* **tue**

 read current;

wenn *understand* **dann**

 go to next section;

 current section becomes this one;

sonst

 go back to the beginning of current section;

Ende

Ende

Xyzxyzxyz xyz xyzxyzxyzxy Zxyzxyzxyzx yzxyzxyz xyz XyzxyzXyzxyzxy. Yzxyzxyz-
zxyz xyz XyzxyzxyzxyZxyzxyzxyz Xyz xyz xyzxy Zxyzxyzxyzxyzxy zxyzxyzxyzx Yzxyz-
zxyzx yzxyzx yzx yzxyzxyzxyzxy Zxyzxyzxyzxyzxyz xyz xyz Xyzxyz Xyzxyzxyzxy zx.

Algorithmus 7.2 : disjoint decomposition

input : A bitmap Im of size $w \times l$

output : A partition of the bitmap

special treatment of the first line;

für $i \leftarrow 2$ **bis** l **tue**

special treatment of the first element of line i ;

für $j \leftarrow 2$ **bis** w **tue**

$\text{left} \leftarrow \text{FindCompress}(Im[i, j - 1]);$

$\text{up} \leftarrow \text{FindCompress}(Im[i - 1, j]);$

$\text{this} \leftarrow \text{FindCompress}(Im[i, j]);$

wenn *left compatible with this* **dann** // $0(\text{left}, \text{this}) == 1$

wenn $\text{left} < \text{this}$ **dann** $\text{Union}(\text{left}, \text{this});$

sonst $\text{Union}(\text{this}, \text{left});$

Ende

wenn *up compatible with this* **dann**

// $0(\text{up}, \text{this}) == 1$

wenn $\text{up} < \text{this}$ **dann** $\text{Union}(\text{up}, \text{this});$

 // this is put under up to keep tree as flat as possible

sonst $\text{Union}(\text{this}, \text{up});$

 // this linked to up

Ende

Ende

für jedes *element e of the line i* **tue** $\text{FindCompress}(p);$

Ende

7.9 Gleichungen

[illegible]

$$\text{var} \hat{\Delta} = \sum_{j=1}^t \sum_{k=j+1}^t \text{var}(\hat{\alpha}_j - \hat{\alpha}_k) = \sum_{j=1}^t \sum_{k=j+1}^t \sigma^2(1/n_j + 1/n_k). \quad (7.1)$$

$Zxyzxyzxyz\ xy\ zxy\ zxyzxyzxyz\ Yzxyzxyzxyzxyz\ (yzxyzxyzZx,\ yzxyzxyzYz\ xyz\ xyZxyz)$
 $xyzxyzxyz\ zxy\ zxyzxyzxyz\ Zxyzxyzxyz\ xyz\ xyzxyzxyzxyz\ Yzxyzxyzxyz\ Xyzxyzxyzxyz\ zxy$
 $xyz\ Xyzxyzxyzxyzxyz\ zxyzxyzxyz\ Yzxyzxyzxyz\ (Yzxyzxyzxyz).$

$$\frac{d}{dx} \arctan(\sin(x^2)) = -2 \frac{\cos(x^2)x}{-2 + (\cos(x^2))^2}$$

$xyzxyz\ xyz\ xyz\ xyzxy\ zxy\ A1, A2, \dots, Aa. \ xyzx\ Yzxyzxyz\ xyz\ xyzxy\ zxyzxyzx\ Yzxy-$
 $zxyzxyzxyz\ xyzxyz\ xyzx.$

$$\left. \begin{aligned} B' &= -\partial \times E, \\ E' &= \partial \times B - 4\pi j, \end{aligned} \right\} \quad \text{Maxwell's equations} \quad (7.2)$$

Yzxyz xzyxzyxzyxzyz Xyzyxzyxzyxzyz yzx yzx yzxyzyxzy Xyzyxzyxzyxzyxzyxzy xy zxyzyxzy xzy xzy xzyzyxzyxzyxzy Xzyzx yzxyz xyz xzyxzyxzyxzyxzyxzy Yzxyzyxzyxzyxzyxzy.

7.10 Definitionen & Hypothesen

$Zxyzxyzxyz\ xy\ xzyxzyxzy\ Zxyz\ (Xyzxyzxy\ ZX)\ yzx\ yzxy\ zx\ yzxyz\ Xyzxyzxyz\ Xyzxyzxy\ xzy\ Zxyzxyz\ xyzxzyxzyxzy\ Zxyzxy\ (Zxyzxyzxy\ ZX)\ yzxyzxy\ (zxyzx\ Yzxyzxy).$

Definition 1 Let f be a function whose derivative exists in every point, then f is a continuous function.

$xyzxyz, yzxyz\ yz\ yzxyzxyz\ xyzxyzxyz\ zxyzxyz\ xyzxyzxyz\ yz\ xy\ Zxyz$

Definition 2 (Pythagorean theorem) *This is a theorem about right triangles and can be summarised in the next equation*

$$x^2 + y^2 = z^2$$

Yzxyzxyzxyz xyz XyzxyzxyzxyzZxyzxyzxyz Xyz xyz xyzxy Zxyzxyzxyzxyzxyz xxyzxyzxyzx Yxyzxyzxyz yxyzxyz yzx yxyzxyzxyzxyzxyz Zxyzxyzxyzxyzxyz xyz xyz Xyzxyz Xyzxyzxyzxyzxyz zx.

Hypotheses 1 *The greater the service orientation, the greater the level of employee outcomes (i.e. organizational commitment, esprit de corps, and job satisfaction).*

Hypotheses 2 (Business Performance) *The greater the service orientation, the better the business performance (i.e. ROA, new accounts opened, and service quality image)*

Yzx Yzxyzyxzyz (XyzxyZxyzyz Xyzxyzyxzy), zxy Zxyzyzyxzyxzyxzyz (Xyzxy-
zXyzxyzyxzyxYzxyzyxzyz) xyz xyz Xyzxyzyxzyxzy (ZxyzXyzxyzyxzyZxyzyzyxzyx)
yzxy zxyz xyzyx.

7.11 To-Do-Notes

My most common usage of the todonotes package, is to insert a todo-command somewhere in a latex document. An example of this usage is the command `\todo{Make a cake}`, which renders like .

Make a cake

It is possible to place a todonote inside the text instead of placing it in the margin, this could be desirable if the text in the note has a considerable length. `\todo[inline]{A todonote placed in the text}`

A todonote placed in the text

The `\listoftodos`-command inserts a list of all the todos in the current document.

Literaturverzeichnis

- [2] Valeriu Codreanu, Damian Podareanu und Vikram Saletore. *Scale out for large minibatch SGD: Residual network training on ImageNet-1K with improved accuracy and reduced time to train*. 2017. arXiv: 1711.04291 [stat.ML].
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit und Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].
- [4] Jean-Christophe Filliâtre und Sylvain Conchon. „Type-safe Modular Hash-consing“. In: *Proceedings of the 2006 Workshop on ML*. ML '06. Portland, Oregon, USA: ACM, 2006, Seiten 12–19. ISBN: 1-59593-483-9. DOI: 10.1145/1159876.1159880. URL: <http://doi.acm.org/10.1145/1159876.1159880>.
- [5] Ricardo Guerrero. *Training Deep Learning Models On multi-GPUs - BBVA Next Technologies*. URL: <https://www.bbvanexttechnologies.com/blogs/training-deep-learning-models-on-multi-gpus/> (besucht am 8. Aug. 2021).
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].
- [9] Chris Richardson. *Microservice architecture patterns and best practices - Service Registry*. 2014. URL: <http://microservices.io/patterns/service-registry.html> (besucht am 3. Nov. 2015).
- [10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg und Li Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. 2015. arXiv: 1409.0575 [cs.CV].
- [11] Florian Schroff, Dmitry Kalenichenko und James Philbin. „FaceNet: A unified embedding for face recognition and clustering“. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). DOI: 10.1109/cvpr.2015.7298682. URL: <http://dx.doi.org/10.1109/CVPR.2015.7298682>.
- [12] Zhong Shao, John H. Reppy und Andrew W. Appel. „Unrolling lists“. In: *SIGPLAN Lisp Pointers VII.3* (Juli 1994), Seiten 185–195. ISSN: 1045-3563. DOI: 10.1145/182590.182453. URL: <http://doi.acm.org/10.1145/182590.182453>.
- [13] *Stanford DAWN Deep Learning Benchmark (DAWNBench) · ImageNet Training*. URL: <https://dawn.cs.stanford.edu/benchmark/ImageNet/train.html> (besucht am 8. Aug. 2021).
- [14] Mingxing Tan und Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: 1905.11946 [cs.LG].

A Anhang

Eins (ohne extra Eintrag im Inhaltsverzeichnis)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Zwei (ohne extra Eintrag im Inhaltsverzeichnis)

Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Drei (ohne extra Eintrag im Inhaltsverzeichnis)

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Vier (ohne extra Eintrag im Inhaltsverzeichnis)

Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Eidesstattliche Erklärung

Hiermit versichere ich, dass meine Bachelorarbeit „Deep Learning zur visuellen Erkennung von Tierarten“ („Deep learning for visual recognition of animal species“) selbständig verfasst wurde und dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt wurden. Diese Aussage trifft auch für alle Implementierungen und Dokumentationen im Rahmen dieses Projektes zu.

Potsdam, den 10. August 2021,

(Minh Kien Nguyen)