



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Kollisionsvermeidung eines fahrbaren Roboters

-

Implementierung und Training eines neuronalen Netzes mittels Transfer-Learning

Name:

Sebastian Richter

Aaron Zielstorff

Matrikelnummer:

572906

567183

Fachbereich: FB1

Studiengang: M. Elektrotechnik

Fachsemester: 2. FS

Fach: MSS5 Computational Intelligence

Dozent: Prof. Dr.-Ing. Steffen Borchers

Abgabe am: 23. September 2022

Inhaltsverzeichnis

1	Einleitung	4
2	Theoretische Grundlagen	5
2.1	Convolutional Neural Networks (CNN)	5
2.1.1	Convolution	6
2.1.2	Pooling	8
2.1.3	Rectification	9
2.1.4	Fully-Connected	9
2.2	AlexNet	10
2.3	Transfer-Learning	12
3	Versuchsaufbau	13
4	Versuchsdurchführung	14
5	Ergebnisse und Fazit	15
6	Ausblick	16
7	Anhang	17
	Literaturverzeichnis	18
	Bücher	18
	Artikel	18
	Online Quellen	18

Abbildungsverzeichnis

1.1	Illustration Jetbot Roboterfahrzeug	4
2.1	Basisstruktur eines CNN	6
2.2	Unterschiedliche Eingangsbilder mit ähnlichen Merkmalen	6
2.3	Anwendung der Faltung mithilfe eines Filters	7
2.4	Filtertypen und Feature-Maps	8
2.5	Anwendung der Zusammenlegung (max. Pooling)	8
2.6	Rectified Linear Unit	9
2.7	Anwendung der Gleichrichtung (Rectification)	9
2.8	Gewichtungen im Fully-Connected-Layer	10
2.9	Schematischer Aufbau des AlexNet	11
2.10	Visualisierung des Transfer-Learning	12

Tabellenverzeichnis

1 Einleitung

Diese Arbeit beschäftigt sich mit der Thematik „**Künstliche Intelligenz**“. Künstliche Intelligenzen sind Forschungsgegenstand der Neuroinformatik. (vgl. [7, Seite 3]). Speziell der Zweig der **künstlichen neuronalen Netze** (engl. Artificial Neural Network ANN) spielt immer mehr eine große Rolle für reale Anwendungsfälle. Prominente Beispiele sind unter anderem das autonome Fahren und die Worterkennung bei Sprachassistenten. (vgl. [5, Seite 15]).

Künstliche neuronale Netzwerke (im Folgenden abgekürzt mit KNN) sind Netze aus künstlichen Neuronen. Dabei handelt es sich um ein Modell nach dem biologischen Vorbild einer Nervenzelle. (vgl. [5, Seiten 136, 137]). Eine spezielle Form der neuronalen Netze sind die **Convolutional Neural Networks** (im Folgenden abgekürzt mit CNN). Der Name resultiert aus der Anwendung der Faltung bei der Nutzung dieses Netzwerktyps. Es handelt sich auch hier um ein von biologischen Prozessen inspiriertes Konzept im Bereich des **maschinellen Lernens**. (vgl. [3, Seite 50]).

In dieser Arbeit soll zunächst theoretisch das Konzept und die Funktionsweise der CNN beleuchtet werden. Dabei wird vor allem auch auf **AlexNet** als beispielhaftes CNN eingegangen, da dieses im späteren Verlauf Anwendung findet. Darauf aufbauend wird im Anschluss das **Transfer-Learning** erklärt. Auch dieses soll verwendet werden.

Abschließend wird die Arbeit mit der Anwendung der behandelten Konzepte an einem praktischen Versuch getestet. Hierzu sollen in Echtzeit Bilddaten von einer Kamera auf einem kleinen Roboterfahrzeug mittels CNN und Transfer-Learning ausgewertet werden, so dass das Gefährt sich kollisionsfrei im Raum fortbewegen kann. Sämtliche Berechnungen als auch die Steuerung des Roboters werden auf einem **Jetson Nano** Mikrocontroller des Unternehmens Nvidia durchgeführt. Dieses ist bekannt für die Produktion von (Hochleistungs-) Grafikkarten (kurz GPU - Graphical Processing Unit). Der Jetson Controller ist ebenfalls mit einer GPU ausgestattet, die es ermöglicht rechenintensive Operationen, wie z. B. die Berechnung von neuronalen Netzen durchzuführen.

Die Abbildung 1.1 zeigt den Jetson Nano Controller auf dem Roboterfahrzeug „*Jetbot*“ inklusive einer kleinen Weitwinkelkamera.

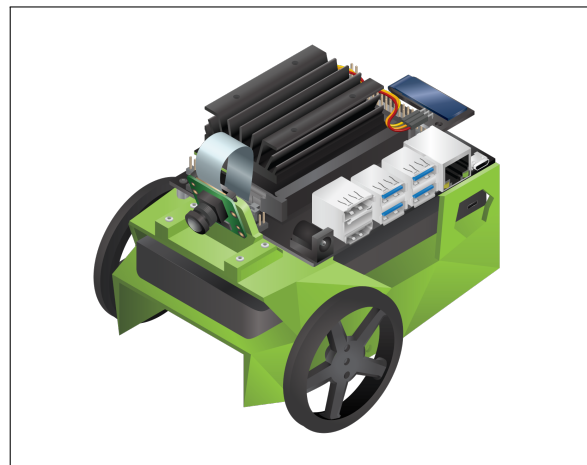


Abb. 1.1: Illustration des Jetbot Roboterfahrzeugs

2 Theoretische Grundlagen

Im ersten Schritt werden die einzelnen Layer eines CNNs mittels Beispielen erläutert. Anschließend erfolgt die Aufbereitung des AlexNets, welches diese Schichten anwendet und als Basis für die Bildverarbeitung zur Kollisionsvermeidung dienen wird. Im letzten Schritt wird auf das Transfer-Learning, sowie dessen Vor- und Nachteile, aber auch dessen Nutzen für die Arbeit eingegangen.

2.1 Convolutional Neural Networks (CNN)

Ein **CNN** ist ein **künstliches neuronales Netz**, welches aus mehreren Schichten (Layern) besteht und Faltungseigenschaften anwendet [15]. Die verschiedenen Layer werden in vier Kategorien eingeteilt, welche in den nachfolgenden Kapiteln erläutert werden.

- Convolution (Faltung)
- Pooling (Zusammenlegung)
- Rectification (Gleichrichtung).
- Fully-Connected (Vollständig verbunden)

Zur schematischen Übersicht des Aufbaus dient Abbildung 2.1. Da der Rectification-Layer nicht immer Anwendung findet, ist dieser nicht in der Abbildung enthalten.

Das CNN wird durch große Datenmengen, deren Eigenschaften und Klassen bekannt sind, vortrainiert. Das Eingangsmedium wird durch unterschiedliche **Filtertypen** auf bestimmte Merkmale untersucht. Die Ergebnisse werden durch **mathematische Operationen** gewichtet und in s.g. **Feature-Maps** gespeichert [15]. Anschließend erfolgt in der **Klassifizierung**, anhand der Gewichtung, die Zuordnung zu einer vorgegebenen Klasse. Die Zuordnung wird anschließend auf Richtigkeit geprüft, um Korrekturen vorzunehmen. Die Eingangsbilder werden mit kleinen Änderungen (Abbildung 2.2) eingegeben, um weitere Feature-Maps zu erzeugen. Dies erfolgt in Form von Feedbackschleifen (Epochen) [15]. Dieser Prozess wird als „**Trainieren des neuronalen Netzes**“ bezeichnet. Wird das trainierte Netz auf unbekannte, jedoch ähnliche Eingangsmedien angewendet, werden die Filter der Merkmale nicht weiter angepasst. Dieser Vorgang heißt „**Anwendung des neuronalen Netzes**“ [15].

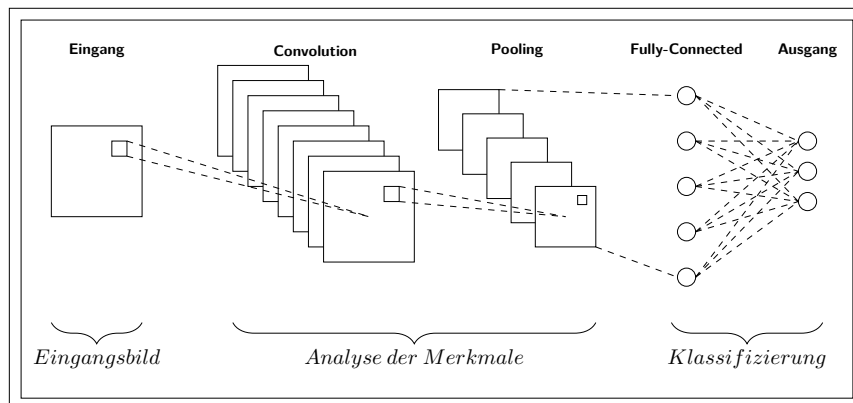


Abb. 2.1: Basisstruktur eines CNN [6]

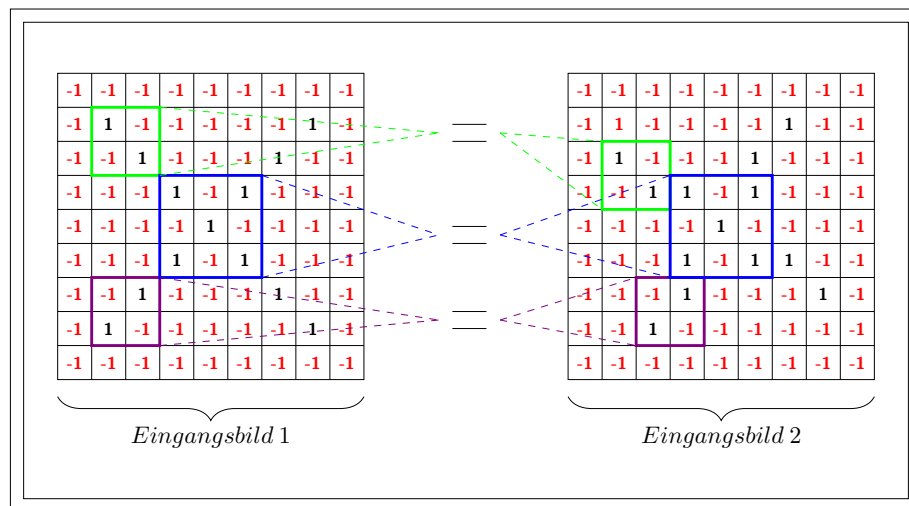


Abb. 2.2: Unterschiedliche Eingangs bilder mit ähnlichen Merkmalen [15]

2.1.1 Convolution

Bei der Faltung werden Filter (Feature) mit **mathematischen Operationen** auf Teilbereiche des Eingangsbildes angewendet. Das Ergebnis der Faltung ist eine **Feature-Map** [15]. Zur Veranschaulichung dient Abbildung 2.3. Der Filter hat beispielhaft eine Größe von 3x3px und wird auf die Teilbereiche im Abstand von 1px angewandt (s. grüne und blaue Markierung im Eingangsbild). Die mathematische Operation gleicht der Multiplikation der einzelnen Werte der Bildpixeln mit den Filterpixeln und der anschließenden **Mittelwertbildung**. Das Ergebnis wird an die jeweilige markierte Stelle in der Feature-Map übernommen [15].

Grüne Markierung:

$$x = \frac{(-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot 1 + (-1) \cdot (-1)}{9} + \frac{(-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot 1}{9}$$

$$x \approx 0.77$$

Blaue Markierung:

$$x = \frac{(-1) \cdot 1 + 1 \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1 + 1 \cdot (-1)}{9} + \frac{(-1) \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1}{9}$$

$$x \approx -0.11$$

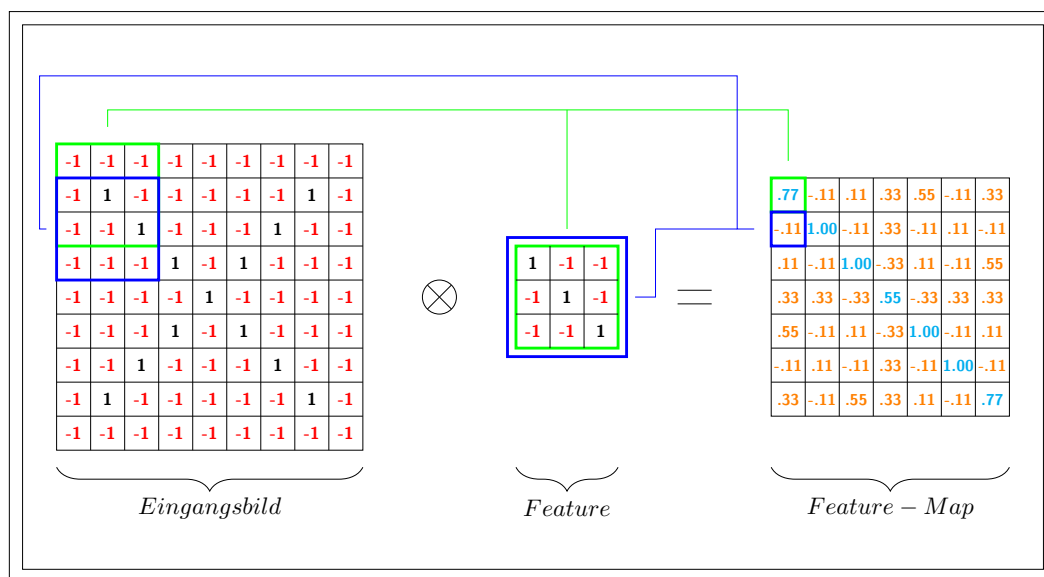


Abb. 2.3: Anwendung der Faltung mithilfe eines Filters [15]

Anhand eines Filters kann keine genaue Auswertung vorgenommen werden. Folglich ist die Anwendung unterschiedlicher Filter notwendig. Jeder Filter erzeugt eine andere Feature-Map, in welcher Auszüge des Originalbilds mit unterschiedlicher Gewichtung erkennbar sind (Abbildung 2.4).

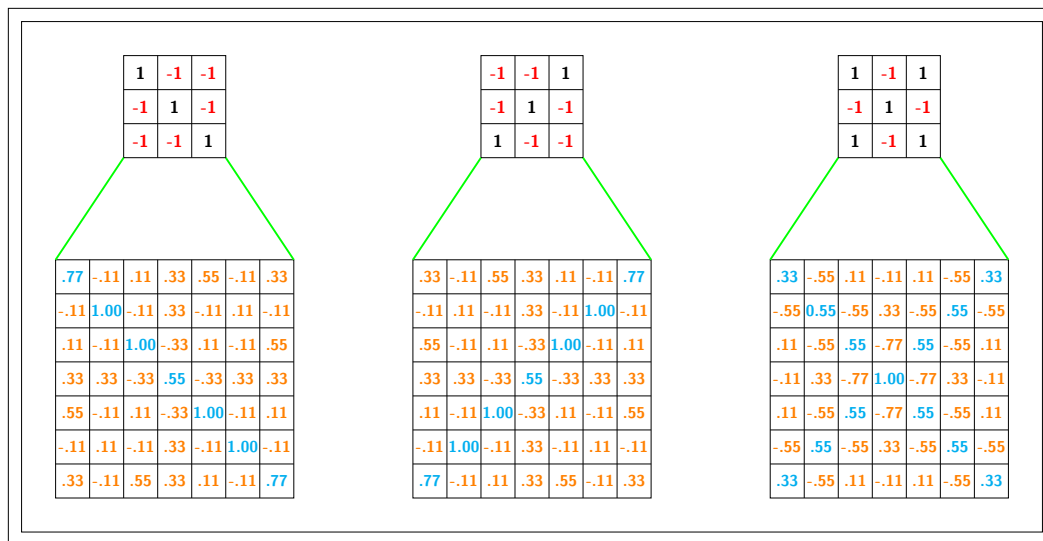


Abb. 2.4: Filtertypen und zugehörige Feature-Maps [15]

2.1.2 Pooling

Beim Maximum Pooling wird in Teilbereichen der Feature-Map nach einem maximalen Wert gesucht. Die Werte werden anschließend zu einer verkleinerten Feature-Map zusammengesetzt. Dies hat den Vorteil, dass Speicherplatz gespart und die höchsten Gewichtungen der Merkmale extrahiert werden können [15]. In Abbildung 2.5 wird exemplarisch eine Fenstergröße von 2x2px mit einer Schrittweite von 2px gewählt.

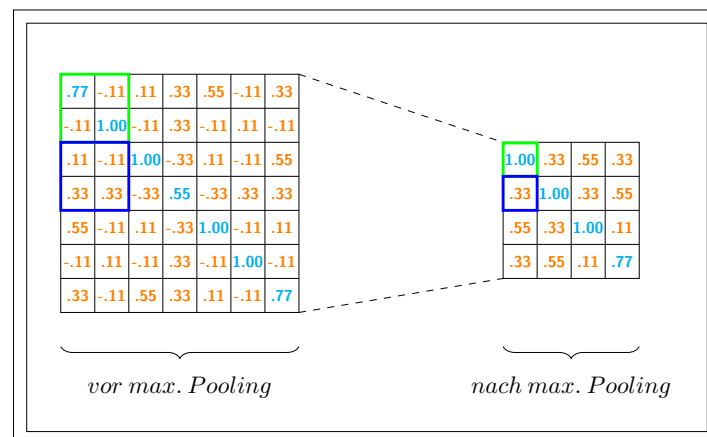


Abb. 2.5: Anwendung der Zusammenlegung (max. Pooling) [15]

2.1.3 Rectification

Im Rectification-Layer werden alle negativen Werte einer Feature-Map zu Null angenommen (Abbildung 2.7). Dieser Prozess reduziert den Rechenaufwand. Zur Anwendung kommt eine lineare Funktion (Rectified Linear Unit (ReLU)) (Abbildung 2.6), die Funktionswerte größer Null unverändert lässt [15].

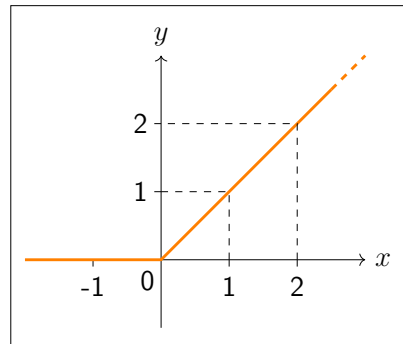


Abb. 2.6: Rectified Linear Unit [15]

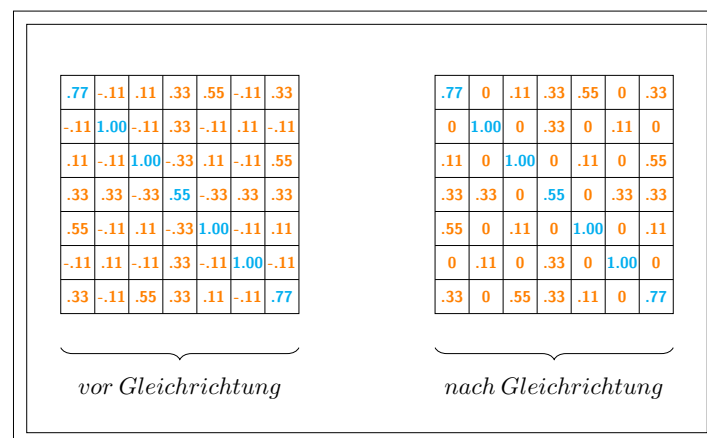


Abb. 2.7: Anwendung der Gleichrichtung (Rectification) [15]

2.1.4 Fully-Connected

Im Fully-Connected-Layer werden die Feature-Maps der vorangegangenen Schicht ausgewertet, um die Wahrscheinlichkeiten der Klassenzugehörigkeiten zu ermitteln [15]. In Abbildung 2.8 werden exemplarisch die fünf größten und fünf kleinsten Gewichtungen gemittelt. Das jeweilige Ergebnis spiegelt die Wahrscheinlichkeit wieder, dass das Eingangsbild

der Klasse „X“ oder „kein X“ zugeordnet werden kann [15].

Wahrscheinlichkeit Klasse „X“:

$$x = \frac{0.90 + 0.87 + 0.96 + 0.89 + 0.94}{5}$$

$$x \approx 0.912 \text{ (91.2\%)}$$

Wahrscheinlichkeit Klasse „kein X“:

$$x = \frac{0.45 + 0.23 + 0.63 + 0.44 + 0.53}{5}$$

$$x \approx 0.456 \text{ (45.6\%)}$$

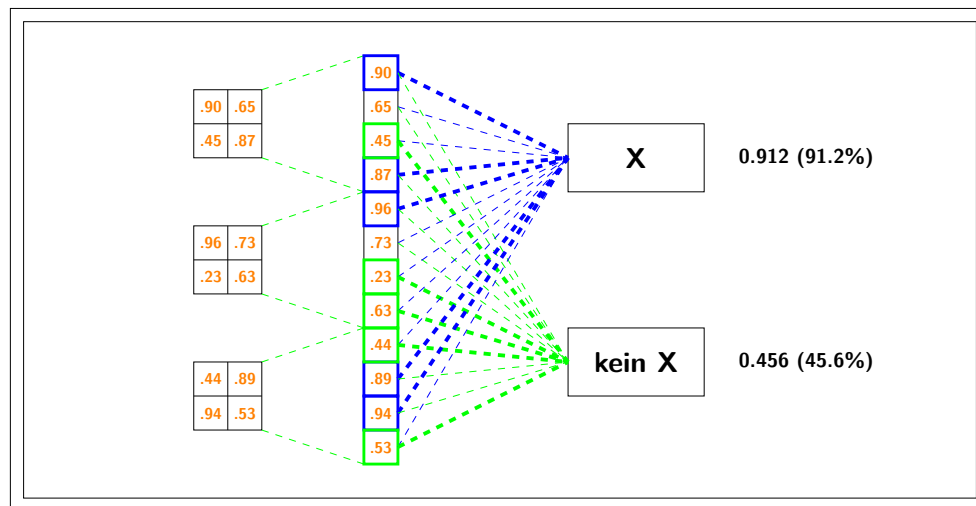


Abb. 2.8: Gewichtungen im Fully-Connected-Layer [15]

2.2 AlexNet

Das AlexNet ist ein Pre-Trained CNN und wurde von Alex Krizhevsky in Zusammenarbeit mit Ilya Sutskever und Geoffrey Hinton entwickelt [12]. Das CNN enthält alle vorher betrachteten Layer-Typen (s. Unterunterabschnitt 2.1.1 bis Unterunterabschnitt 2.1.4).

Das AlexNet wurde 2006 mit 1.2 Mio. Bildern mit einer Größe von jeweils 227x227x3 trainiert und erreichte eine Fehlerrate von 16.4% [2]. Das CNN ermöglichte einfachere und schnellere Fortschritte in der Gesichtserkennung. Mittlerweile wurde das CNN durch das leistungsstärkere GoogleLeNet mit 22 Layern abgelöst.

Das AlexNet besteht aus fünf Convolution- und drei Pooling-Schichten gefolgt von drei Fully-Connected-Layern. Zwischen den Schichten wird teilweise eine ReLU angewendet [10]. Die schematische Struktur ist in Abbildung 2.9 dargestellt.

Das AlexNet wird aufgrund des einfachen Aufbaus im Weiteren zur Bildverarbeitung zur Vermeidung von Kollisionen verwendet.

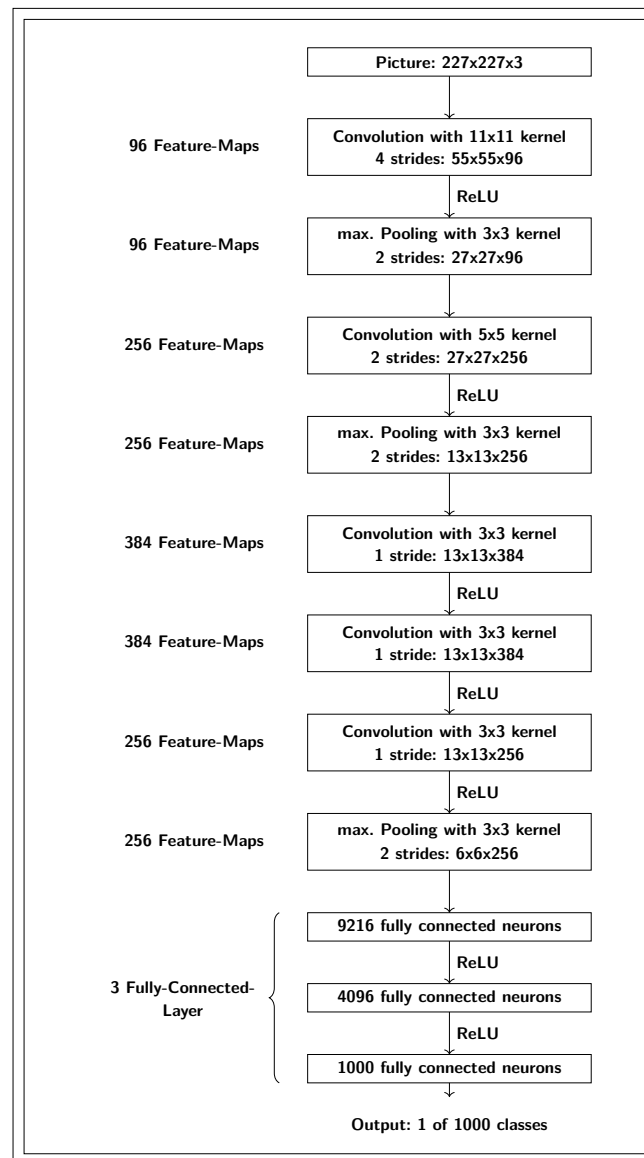


Abb. 2.9: Schematischer Aufbau des AlexNet [12]

2.3 Transfer-Learning

Um das AlexNet aus Unterabschnitt 2.2 sinnvoll nutzen zu können, wird die Maschine-Learning-Technik des **Transfer-Learning** angewandt. Beim Transfer-Learning wird ein bereits **vortrainiertes neuronales Netz** (z.B. CNN oder ResNet) auf ein **ähnliches Problem** angewendet (z.B. Bild- und Textverarbeitung) [13]. Die Anwendung dessen ist dann sinnvoll, wenn nur ein kleiner eigener Datensatz mit wenigen Klassen zur Verfügung steht. Durch den Transfer werden sämtliche Skills und Eigenschaften übernommen und auf das eigene Problem adaptiert (Abbildung 2.10). Dies reduziert den eigenen Ressourceneinsatz, ermöglicht eine schnellere Erstellung und Anwendung auf ein Problem und erhöht die Modellqualität [13].

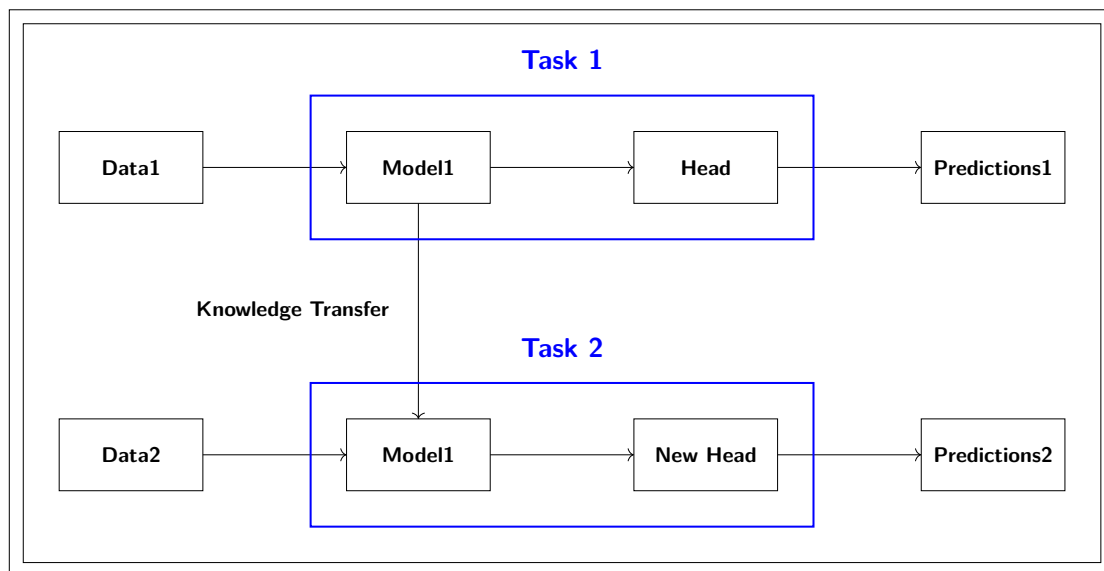


Abb. 2.10: Visualisierung des Transfer-Learning [9]

Das Transfer-Learning wird in die Bereiche **Pre-Training** und **Fine-Tuning** unterschieden. Beim Pre-Training werden die Feature-Maps des neuronalen Netzes unverändert übernommen und lediglich die Zuordnung der Klassen (Predictions) an die Klassen des neuen Problems angepasst und trainiert [13].

Beim Fine-Tuning werden zusätzlich die Feature-Maps entsprechend auf das spezifische Problem angepasst. Hierzu werden größere eigene Datenmengen benötigt [13].

Aufgrund des geringen eigenen Datensatzes wird das Pre-Training des AlexNets bevorzugt angewendet.

3 Versuchsaufbau

4 Versuchsdurchführung

5 Ergebnisse und Fazit

6 Ausblick

7 Anhang

Literaturverzeichnis

Bücher

- [1] T. Amaratunga, *Deep Learning on Windows*. Apress, Dez. 2020, Kapitel Transfer Learning, 338 Seiten, ISBN: 978-1-4842-6431-7. Adresse: https://www.ebook.de/de/product/41243234/thimira_amaratunga_deep_learning_on_windows.html.
- [2] P. Kalaiarasi und P. E. Rani, *Advances in Smart System Technologies*. Springer-Verlag GmbH, Aug. 2020, Kapitel A Comparative Analysis of AlexNet and GoogLeNet with a Simple DCNN for Face Recognition, 836 Seiten, ISBN: 978-981-15-5029-4. Adresse: https://www.ebook.de/de/product/39551333/advances_in_smart_system_technologies.html (siehe Seite 10).
- [3] K. B. Prakash, Y. V. R. Nagapawan und G. R. Kanagachidambaresan, *Programming with TensorFlow*. Springer International Publishing, Jan. 2021, Kapitel Convolutional Neural Networks, 190 Seiten, Convolutional Neural Networks, ISBN: 978-3-030-57077-4. Adresse: https://www.ebook.de/de/product/41247357/programming_with_tensorflow.html (siehe Seite 4).
- [5] Z. A. Styczynski, K. Rudion und A. Naumann, *Einführung in Expertensysteme*. Springer-Verlag GmbH, Mai 2017, 250 Seiten, ISBN: 9783662531723. Adresse: https://www.ebook.de/de/product/29262042/zbigniew_a_styczynski_krzysztof_rudion_andre_naumann_einfuehrung_in_expertensysteme.html (siehe Seite 4).

Artikel

- [6] P. D.-I. S. Borchers-Tigasson, „Summary and Remarks on CNN,“ *Computational Intelligence*, 2022 (siehe Seite 6).
- [7] U. Schäffer und J. Weber, „Künstliche Intelligenz,“ *Controlling Management Review*, Jahrgang 65, Nummer 2, Seiten 3–3, Feb. 2021. DOI: 10.1007/s12176-021-0370-0 (siehe Seite 4).

Online Quellen

- [8] A. Regensburg. „Hochschullogo HTW Berlin.“ (Juli 2018), Adresse: https://www.akademieregensburg.de/files/akademie/img/hochschulen_logos/18.jpg (besucht am 25. 08. 2022).

- [9] S. Sonwane. „Transfer Learning From Pre-Trained Model for Image Recognition.“ (Mai 2020), Adresse: <https://sagarsonwane230797.medium.com/transfer-learning-from-pre-trained-model-for-image-facial-recognition-8b0c2038d5f0> (besucht am 19.09.2022) (siehe Seite 12).
- [10] S. Vasudevan. „10. AlexNet - CNN Explained and Implemented.“ (Apr. 2020), Adresse: <https://www.youtube.com/watch?v=8GheVe2UmUM> (besucht am 19.09.2022) (siehe Seite 11).
- [11] Wikimedia. „Logo HTW Berlin.“ (Okt. 2014), Adresse: https://upload.wikimedia.org/wikipedia/commons/7/7e/Logo_HTW_Berlin.svg (besucht am 25.08.2022).
- [12] Wikipedia. „AlexNet.“ (Sep. 2022), Adresse: <https://en.wikipedia.org/wiki/AlexNet> (besucht am 19.09.2022) (siehe Seiten 10, 11).
- [13] F. Woelki. „Was ist das Transfer Learning? | Künstliche Intelligenz.“ (Feb. 2020), Adresse: https://www.youtube.com/watch?v=K_csnXsNN5Q (besucht am 19.09.2022) (siehe Seite 12).
- [14] L. Wuttke. „Transfer Learning: Grundlagen und Einsatzgebiete.“ (), Adresse: <https://datasolut.com/was-ist-transfer-learning/> (besucht am 19.09.2022).
- [15] U. Würtz. „CNNs Convolutional Neural Networks Basiswissen.“ (Dez. 2019), Adresse: <https://www.youtube.com/watch?v=OV0KXyYpEZY> (besucht am 19.09.2022) (siehe Seiten 5–10).