

---

# Machine Learning for Computer Vision

## Abschlussprojekt: Rindenklassifizierung

---

**Paul Walker**

Department of Computer Science  
DHBW Stuttgart  
inf20045@dhw-stuttgart.de

**Tom Hofer**

Department of Computer Science  
DHBW Stuttgart  
inf20173@lehre-dhw-stuttgart.de

### Abstract

Diese Arbeit befasst sich mit der Entwicklung eines Machine Learning Modells zur Klassifikation von Rinde zu bestimmten Baumarten. Das Modell könnte in der Praxis beispielsweise in einer App zur Bestimmung der Baumart anhand eines Fotos der Baumrinde eingesetzt werden. Im Rahmen der Arbeit werden vier Modelle basierend auf den vortrainierten Modellen InceptionV3, MobileNetV2, VGG16 und VGG19 erstellt, getestet und gegenübergestellt. Für das Training der Modelle kommen ein Trainingsset mit Daten aus dem Internet und ein Validationset mit selbst erhobenen Daten zum Einsatz.

## 1 Einleitung

Unser Abschlussprojekt befasst sich mit der Entwicklung eines Machine Learning Modells zur Bestimmung der Baumart anhand eines Fotos der Baumrinde. Die Rinde eignet sich sehr gut für die Bestimmung der Baumart, da diese, anders als zum Beispiel Blätter oder Früchte, unabhängig von der Jahreszeit ist. Die Eingabe für unseren Algorithmus ist also ein Bild. Zur Klassifizierung der Baumart verwenden wir ein vortrainiertes Convolutional Neural Network, welches wir für unseren Anwendungsfall anpassen. Die Ausgabe, beziehungsweise Vorhersage unseres Algorithmus ist die Art des Baumes, dessen Rinde auf dem Foto abgebildet ist.

In der Praxis könnte ein solches Modell zum Beispiel in einer App eingesetzt werden, die Baumarten bestimmen soll. Das Erkennen der Baumart anhand der Rinde ist ein schwieriges Problem, da es teilweise große Varianzen innerhalb einer Klasse und kleine klassenübergreifende Varianzen gibt. Abbildung 1 zeigt eine Kiefer die sehr ähnlich zu einer Lärche aussieht und eine weitere Lärche die sich von den anderen beiden Bäumen relativ stark unterscheidet.

## 2 Stand der Technik

Die Rindenklassifikation ist kein neues Problem und es wurden bereits Ansätze sowohl mit klassischer Computer Vision, als auch mit Hilfe von Deep Learning entwickelt. [1]

## 3 Daten und Features

Unser Trainingsdatenset besteht aus 21491 Bildern von Rinde aus dem Internet. Die Daten stammen aus dem BARK-KR [2] Datenset, dem Tree Species Dataset [3] und dem BarkNet 1.0 [4] Datenset. Einige Beispiele aus den Trainingsdaten sind in Abbildung 2 abgebildet. Für das Validationset haben

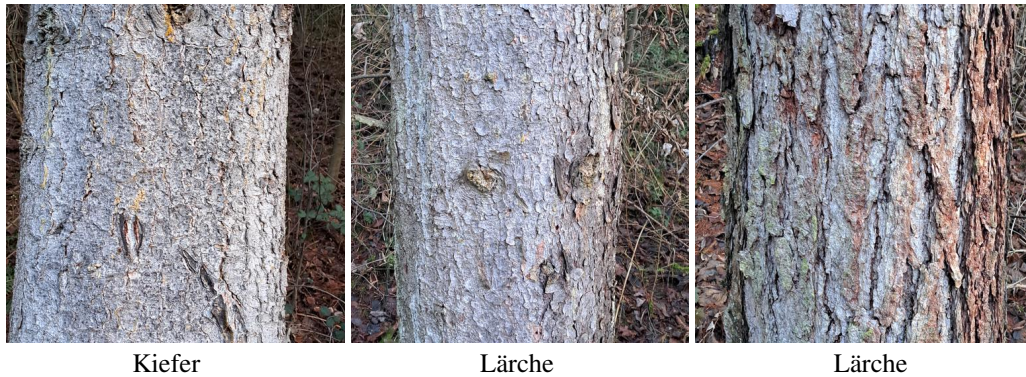


Abbildung 1: Schwierigkeit des Problems

wir 333 eigene Fotos aufgenommen. Einige Beispiele sind in Abbildung 3 abgebildet. Ein Testset haben wir nicht, wir können also keine genaue Aussage über die tatsächliche Feldperformance unseres Modells treffen. Alle Bilder werden auf eine Größe von 512\*512 Pixeln gebracht. Um Verzerrungen zu vermeiden werden die Grafiken zunächst durch Zuschneiden auf das richtige Seitenverhältnis gebracht. In unserem Validationset haben wir acht Klassen, beziehungsweise acht Baumarten. Wir nutzen von den Trainingsdaten nur die Arten, die auch im Validationset vorkommen. Zusätzlich wurden die Label der Trainingsdaten angepasst, da diese die Bäume noch in genaue unterarten einteilen, wir interessieren uns jedoch nur für die Überart. So wird beispielsweise die Orientalische Weiß-Eiche *Quercus aliena* in unserem Datensatz einfach als Oak, also Eiche, klassifiziert. Die vertretenen Baumarten sind Esche, Buche, Birke, Tanne, Lärche, Eiche, Kiefer und Fichte.

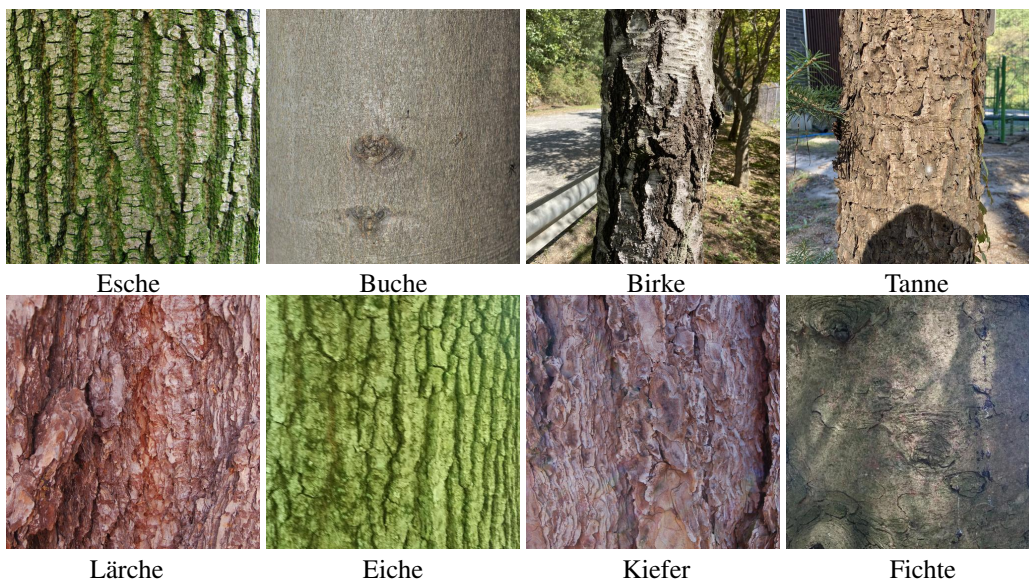


Abbildung 2: Beispiele aus den Trainingsdaten

## 4 Methoden

Zur Erstellung unseres Modells verwenden wir Transfer learning. Das bedeutet wir verwenden ein bereits für Bilddaten Trainiertes Modell aus dem Internet und passen dieses auf unseren Anwendungsfall an. Die Anpassung erfolgt in den Ersten und letzten Schichten des Modells. Die erste Schicht wird auf unsere Eingabe, ein Bild mit 512\*512 Pixeln und drei Kanälen, angepasst. Diese hinzugefügten Schichten sind beispielhaft in dem in Abbildung 4 dargestellten, auf VGG16 basierenden Modell, grün markiert. Außerdem wird eine zweite Schicht eingefügt die ein Rescaling der Wertebereiche von



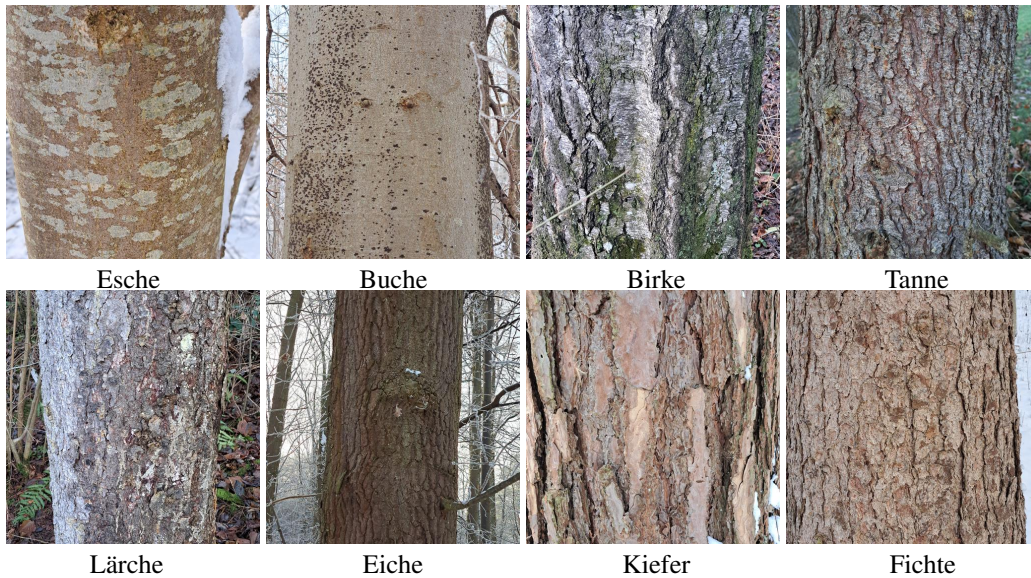


Abbildung 3: Beispiele aus dem Validationset

0 bis 255 auf einen Wertebereich von 0 bis 1 transformiert. Für die Ausgabe des Modells werden die letzten vier Schichten angepasst. Zunächst erfolgt ein zeidimensionales global average pooling gefolgt von einer flatten und dense Schicht. Die dense Schicht verwendet eine Relu als Aktivierungsfunktion. In der letzten Schicht wird die Ausgabe des Modells nochmal auf die Anzahl an Klassen, also in unserem Fall acht, heruntergebrochen. Die Letzte Schicht ist ein Softmax-Layer. Diese Schichten sind in der Abbildung 4 rot markiert.

Wir verwenden für unser Modell einen sparse categorical crossentropy loss.

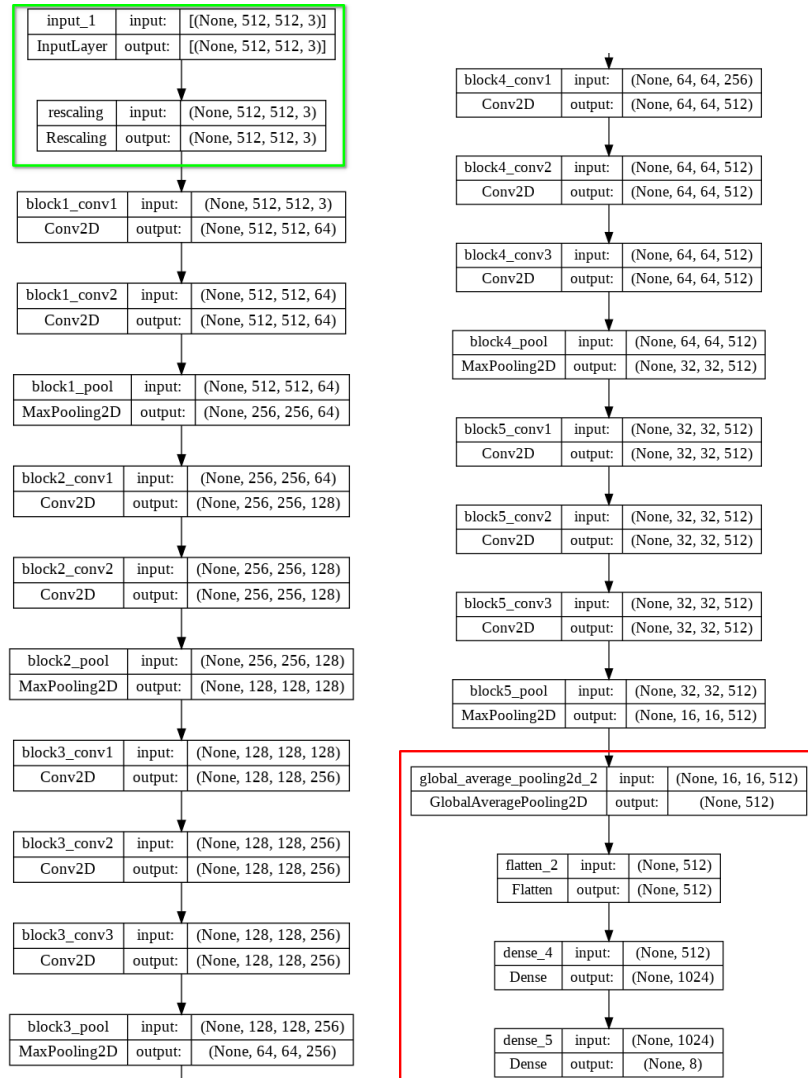


Abbildung 4: Darstellung des auf VGG16 basierenden Modells

## 5 Experimente/Ergebnisse/Diskussion

Wir haben verschiedene vortrainierte Basismodelle ausprobiert und deren Performance gegenübergestellt. Die betrachteten vortrainierten Modelle sind InceptionV3, MobileNetV2, VGG16 und VGG19. Die Ergebnisse der Gegenüberstellung sind in Abbildung 5 dargestellt. Die höchste Performance erzielt MobileNetV2 mit einer Trefferate von 52,85% auf dem Devset mit unseren selbst gelabelten Daten. Bei acht Klassen würde man mit Raten auf eine Trefferate von 12,5% kommen, das Modell ist also deutlich besser als Raten. Da alle Modelle jedoch auf unseren Daten eine sehr viel geringere Trefferchance haben als beim Training, haben wir ein weiteres Devset mit Daten aus dem Internet erstellt. Auf diesen Daten weisen die Modelle eine ähnliche Trefferate wie beim Training auf. Unsere Modelle können also nicht gut von den Trainingsdaten auf unsere eigenen Daten generalisieren. Wir vermuten die Ursache in den regional verschiedenen Unterarten. Um ein besseres Ergebnis zu erzielen müssten mehr regionale Daten erhoben werden, die dem Trainingsset hinzugefügt werden.

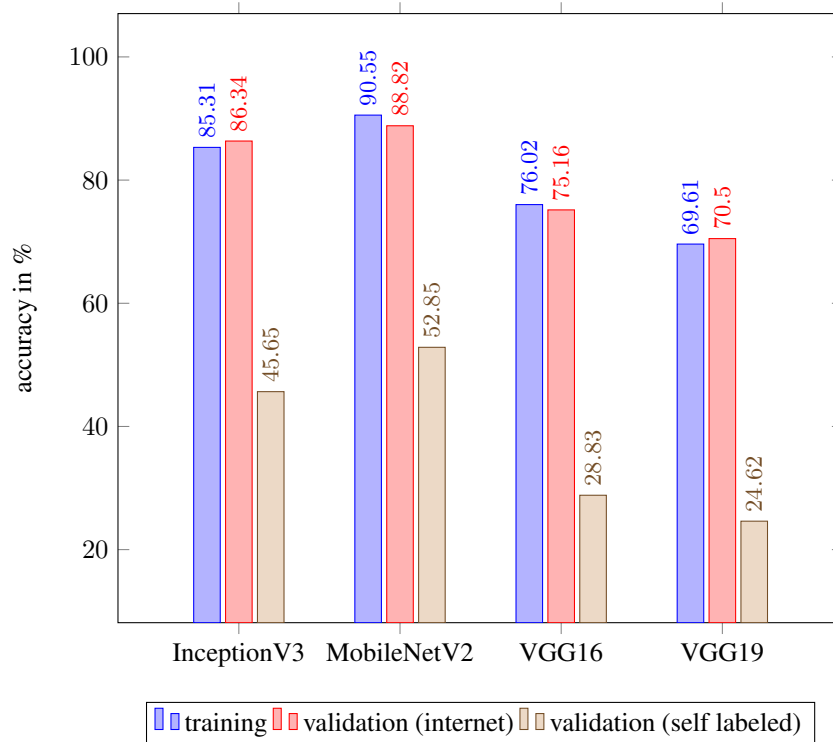


Abbildung 5: Vergleich der verschiedenen Modelle

## 6 Fazit und Ausblick

In dieser Arbeit wurden vier Neuronale Netze mittels Transfer Learning aus vier vortrainierten Modellen erstellt. Diese wurden mit Bildern von Rinde aus dem Internet trainiert, mit dem Ziel der Klassifikation zu einer von acht Baumarten. Zusätzlich wurden händisch Daten erhoben und gelabelt, die das Validationset bildeten. Die vier entwickelten Modelle wurden nach ihrer Treffwahrscheinlichkeit gegenübergestellt. AuSSerdem wurde eine schlechte Generalisierungsfähigkeit der Modelle von den externen auf die selbst erstellten Daten festgestellt.

Da die Performance auf einem aus den externen Daten erstellten Validationset sehr nahe an der Performance auf den Trainingsdaten liegt, kann davon ausgegangen werden, dass bei einem Training des Modells mit regionalen Daten eine ähnlich hohe Performance erreicht werden kann.

## Referenzen

### Verwendete Softwarebibliotheken

- Pillow-SIMD
- google.colab (drive)
- Tensorflow
- os
- Keras
- matplotlib

[1] Misra, D., Crispim-Junior, C., Tougne, L. (2020). Patch-Based CNN Evaluation for Bark Classification. In: Bartoli, A., Fusiello, A. (eds) Computer Vision ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science(), vol 12540. Springer, Cham. [https://doi.org/10.1007/978-3-030-65414-6\\_15](https://doi.org/10.1007/978-3-030-65414-6_15)

- [2] Fiel, Stefan, & Sablatnig, Robert. (2010). Tree Species Dataset consisting of Images of the Bark, Leaves or Needles [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.4446955>
- [3] Fiel, Stefan, & Sablatnig, Robert. (2010). Tree Species Dataset consisting of Images of the Bark, Leaves or Needles [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.4446955>
- [4] Giguère, Philippe; Carpentier, Mathieu; Gaudreault, Jonathan (2019), BarkNet 1.0 (Part 1 of 4), Mendeley Data, V1, doi:10.17632/zgr7r2r4nt.1