
Machine Learning for Computer Vision

Abschlussprojekt: Rindenklassifizierung

Paul Walker

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

Tom Hofer

Department of Computer Science
DHBW Stuttgart
inf20173@lehre-dhbw-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.

2 Stand der Technik

https://link.springer.com/chapter/10.1007/978-3-030-65414-6_15

3 Daten und Features

Unser Trainingsdatenset besteht aus 21491 Bildern von Rinde aus dem Internet. Diese beziehen wir von den folgenden Quellen: <https://zenodo.org/record/4872489>, <https://zenodo.org/record/4446955> und <https://data.mendeley.com/datasets/zgr7r2r4nt/1>. Einige Beispiele aus den Trainingsdaten sind in Abbildung 1 abgebildet. Für das Validationset haben wir 333 eigene Fotos aufgenommen. Einige Beispiele sind in Abbildung 2 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. Die vertretenen Baumarten sind Esche, Buche, Birke, Tanne, Lärche, Eiche, Kiefer und Fichte.

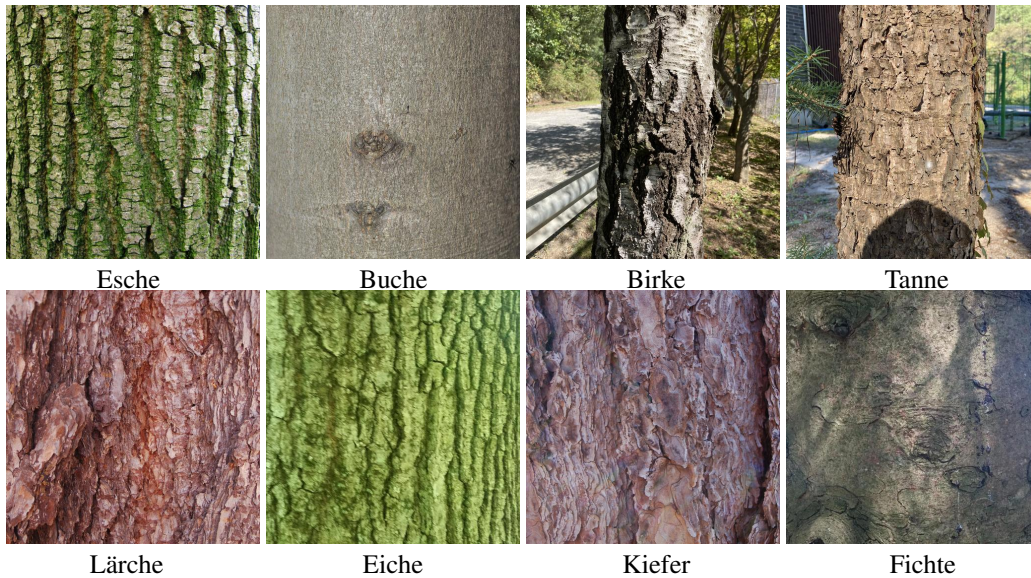


Abbildung 1: Beispiele aus den Trainingsdaten

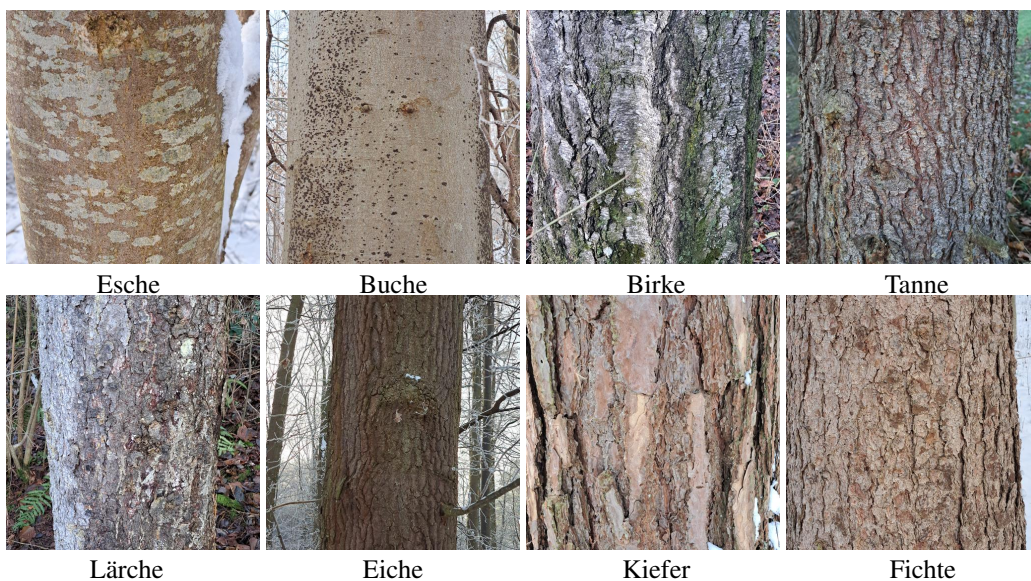


Abbildung 2: Beispiele aus dem Validationset

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 3 dargestellten, auf VGG16 basierenden Modell, grün markiert. Außerdem wird eine zweite Schicht eingefügt die ein Rescaling der Wertebereiche von 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 3 rot markiert.

Wir verwenden für unser Modell einen sparse categorical crossentropy loss.

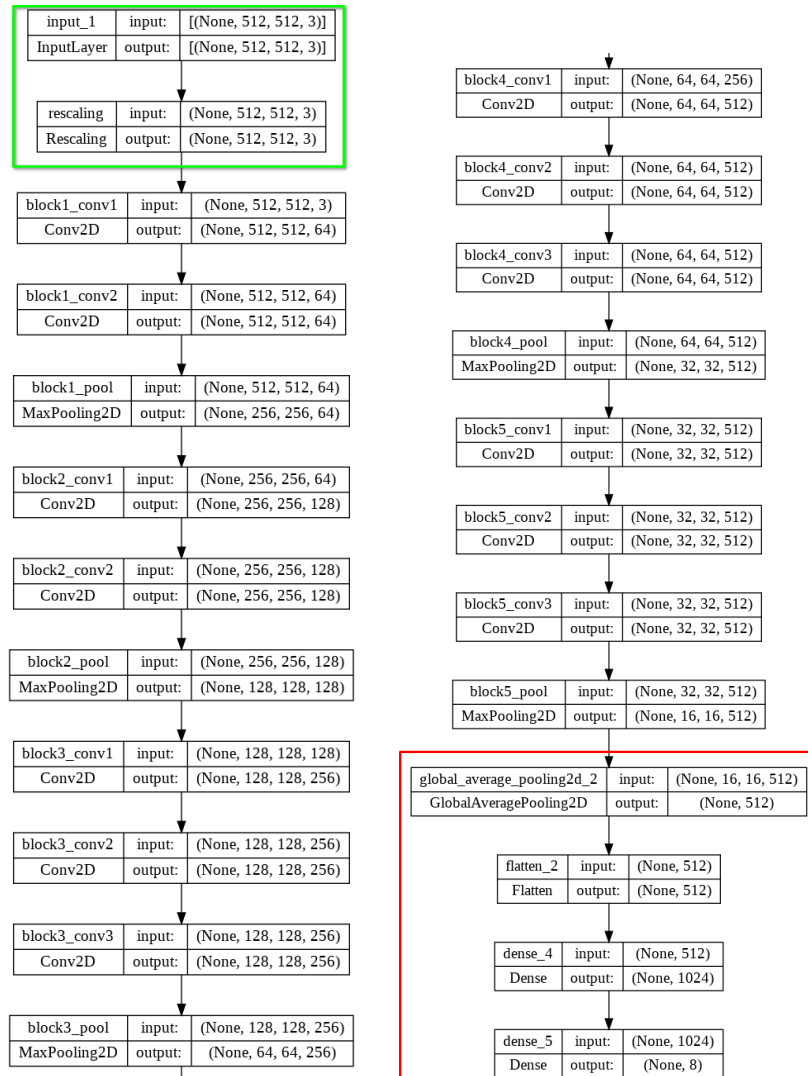


Abbildung 3: Darstellung des auf VGG16 basierenden Modells

5 Experimente/Ergebnisse/Diskussion

Wir haben verschiedene vortrainierte Basismodelle ausprobiert und deren Performance gegenübergestellt. Die betrachteten vortrainierten Modelle sind InceptionV3, MobileNetV2, VGG16 und VGG19.

6 Fazit und Ausblick

Referenzen

Libraries used

- Pillow-SIMD
- google.colab (drive)

- Tensorflow
- os
- Keras
- matplotlib