Machine Learning Projekt

Bildverarbeitung: Klassifizierung Quallenarten

Studentin/Student *Simon Kohler  
Benedikt Sele*

Auftraggeberin *Martin Pischtschan*

ModulGrundlagenlabor 4 – Machine Learning

Fachhochschule Nordwestschweiz, Hochschule für Technik

*Windisch, März 2025*

Inhaltsverzeichnis

[Abbildungsverzeichnis iii](#_Toc193026573)

[Tabellenverzeichnis iii](#_Toc193026574)

[1 Einleitung 1](#_Toc193026575)

[2 Grundlagen 2](#_Toc193026576)

[2.1 Datensatz 2](#_Toc193026577)

[2.2 Unterkapitel 2](#_Toc193026578)

[3 Titel 3](#_Toc193026579)

[3.1 Untertitel 1 3](#_Toc193026580)

[3.2 Untertitel 2 3](#_Toc193026581)

[3.2.1 Untertitel 3. Grades 3](#_Toc193026582)

[4 Titel 4](#_Toc193026583)

[5 Schlussbemerkungen 5](#_Toc193026584)

[Quellenverzeichnis 6](#_Toc193026585)

[Anhang 7](#_Toc193026586)

Abbildungsverzeichnis

No table of figures entries found.

Tabellenverzeichnis

**No table of figures entries found.**

# Einleitung

# Grundlagen

## Datensatz

## Unterkapitel

# Vorgehensweise

## Strukturierung des Datensatzes

Convert\_jpegs\_to\_numpy.py

Da die gängigen ML-Modelle als Eingabe einheitliche, numerische Eingangsgrössen erwarten, werden alle Bilder des Datensatzes als erstes zur Weiterverarbeitung skaliert und in NumPy-Arrays konvertiert.

Eine einheitliche Eingangsgrösse wird erreicht, indem die Bilder des Datensatzes auf ihre Grösse (Pixel x Pixel) geprüft und falls abweichend per Lanczos-Filter auf 179x179 Pixel skaliert werden.

Die skalierten Bilder werden anschliessend mittels np.asarray() in ein mehrdimensionales numerisches Array umgewandelt. Das entstehende Array hat die Form (179, 179, 3) was der Höhe, Breite und der Anzahl Farbkanäle (RGB = 3) entspricht.

Load\_jellyfish\_data.py

Die so erstellten .npy-Files werden anschliessend geladen und mithilfe ihres Speicherorts (Ordnernahme = Quallenart) einer Kategorie zugeordnet. Die insgesamt sechs zu unterscheidenden Quallenarten werden nummerisch kategorisiert und jeweils als ID den NumPy-Arrays hinzugefügt. Dadurch entsteht ein strukturiertes Datenset aus Bilddaten und Klassenlabels, welches die Grundlage für das Training und die Evaluation unseres ML-Modells bildet.

## Datenaufbereitung für das Modell

Training.py

Als Erstes werden die Pixelwerte der als Numpy-Array vorliegenden Bilder normalisiert. Das heisst, die als im Bereich von 0 bis 255 vorliegenden RGB-Farbkanalwerte (z.B. 120, 200, 50) werden durch 255 geteilt und somit auf den Bereich 0 bis 1 skaliert. Dies ermöglicht stabile Berechnungen, verhindert extreme Gradienten Änderungen und verbessert die Konvergenz des Modells, wodurch das Lernen effizientes wird.

Anschliessend werden die als Zahlen gespeicherten Klassenlabels (Quallenarten) mittels One-Hot-Encoding in Vektoren umgewandelt. Dies verhindert eine falsche numerische Ordnung der Klassen und ermöglicht eine korrekte Wahrscheinlichkeitsverteilung im Modell.

## Modellarchitektur

Das verwendete Concolutional Neural Network (CNN) verwrbeitet die als (179, 179, 3) vorliegenden Bilddaten schrittweise durch mehrere Schichten:

1. Erste Faltungsschicht Conv2D (grob)
   1. Erkennt niedrigstufige Merkmale wie Kanten, Linien oder einfache Strukturen
2. Erste Max-Pooling Schicht
   1. Nur die stärksten Werte werden übernommen (2x2)
   2. Wichtigste Information bleibt erhalten während die Anzahl Berechnungen reduziert wird
3. Zweite Faltungschicht Conv2D mit 64 Filtern
   1. Erkennt höherwertige Muster, Formen, Texturen oder charakteristische Merkmale der Quallenarten
4. Zweite Max-Pooling Schicht
   1. Reduziert die Rechenlast
5. Flatten-Schicht
   1. Wandelt die als mehrdimensionale Matrix vorliegenden Informationen in einen eindimensionalen Vektor um.
   2. Dient als Eingabe für die Klassifizierungsschicht
6. Dropout-Schicht
   1. Vermeiden von Overfitting
   2. Während des Trainings werden zufällig 50% der Neuronen deaktiviert
7. Dense-Schicht
   1. Schicht mit Ausgangsneuronen

## Optimierung

# Ergebnisse

# Diskussion

Quellenverzeichnis

Nach APA:

Cherny, B. (2019). *Programming TypeScript: Making your javascript applications scale* (First edition). O’Reilly Media.

Hering, E., Martin, R. & Stohrer, M. (2016). *Physik für Ingenieure* (12. Auflage). Springer Vieweg.

Scholz, R. (2018). *Grundlagen der Elektrotechnik: Eine Einführung in die Gleich- und Wechselstromtechnik*. Fachbuchverlag Leipzig im Carl Hanser Verlag. https://doi.org/10.3139/9783446456310

Nach IEEE:

[1] B. Cherny, *Programming TypeScript: making your javascript applications scale*, First edition. Sebastopol CA: O’Reilly Media, 2019.

[2] E. Hering, R. Martin und M. Stohrer, *Physik für Ingenieure*, 12. Auflage. Berlin: Springer Vieweg, 2016.

[3] R. Scholz, *Grundlagen der Elektrotechnik: eine Einführung in die Gleich- und Wechselstromtechnik*. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2018.

**Anhang**