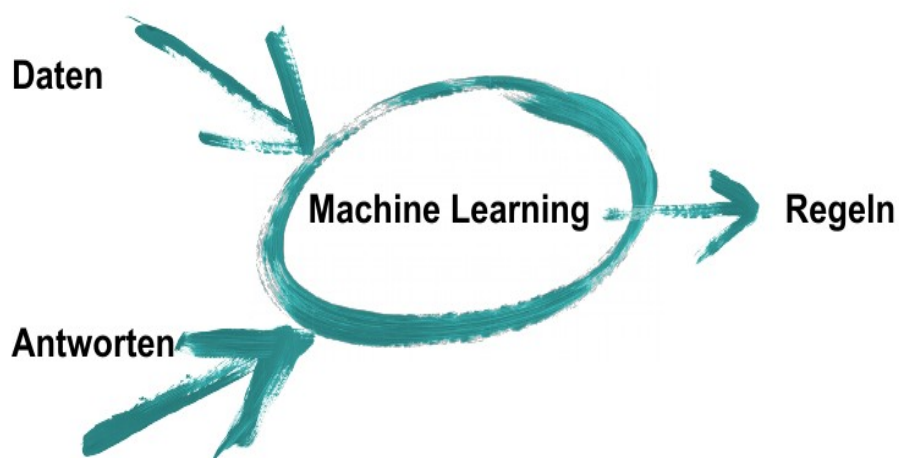


KI programmieren im Informatikunterricht Teil 2: Bilderkennung



[B0]

Abb. [B0]: „E-V-A“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Unterricht"



Inhaltsverzeichnis

KI programmieren im Informatikunterricht Teil 2: Bilderkennung.....	1
A Übersicht.....	3
A 1 Zusammenfassung:.....	3
A2 Stundenübersicht.....	4
A3 Themeneinstieg und theoretische Grundlagen.....	5
B Lernaufgabe.....	6
B1 Unterrichtsbegleitende Präsentation.....	6
B2 Arbeitsblätter.....	17
C Bezug zum Rahmenlehrplan.....	20
C1 Didaktischer Kommentar.....	20
C2 Bezüge zum Rahmenlehrplan Informatik.....	21
C3 Bezüge zum Basiscurriculum Medienbildung.....	23
C4 Bezüge zu übergreifenden Themen.....	23
C5 Bezüge zu anderen Fächern:.....	24
D Anhang.....	25
D1 „Material“ für den Einsatz dieser Lernaufgabe.....	25
D2 Hinweise / Musterlösung der Lernaufgabe.....	25
D3 Quellen / Lektüreliste zum Weiterlesen.....	28
D4 Bildnachweise.....	29



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bilderkennung

Senatsverwaltung
für Bildung, Jugend
und Familie:

BERLIN



A Übersicht

A 1 Zusammenfassung:

Wer neuronale Netze einmal selbst trainiert hat, kann deren Risiken, Probleme und Chancen und damit auch mögliche gesellschaftliche Entwicklungen besser einschätzen.

In dieser Lernaufgabe lernen die Schüler:innen den Aufbau von neuronalen Netzen (NN) mithilfe von TensorFlow so, dass Bilder erkannt werden können. Im Anschluss ist es den Schüler:innen möglich, Modelle in TensorFlow für eigene Projekte zu entwerfen und zu trainieren.

Diese Lernaufgabe setzt die Grundlagen der ersten Lernaufgabe: „KI programmieren im Informatikunterricht Teil I: Einführung“ voraus. Ebenso wie Grundlagen der Programmierung mit Python, das Einbinden von Bibliotheken und den Umgang mit einer entsprechend leistungsfähigen IDE wie z. B. Spyder.

In diesem Sinne: *All hands on code!*

Intention der Lernaufgabe

- Kennenlernen der Grundlagen von *computer vision*
- Kennenlernen / Erstellung / Programmierung eines eigenen NN mithilfe von TensorFlow zur Bilderkennung
- Erkennen von Chancen und Risiken und möglichen gesellschaftlicher Auswirkungen durch *computer vision*.

Hinweise:

Das Zusammenspiel von TensorFlow und Python läuft nicht immer problemlos. Bei unklaren Schwierigkeiten hilft es, eine eher ältere Python-Version mit einer eher neueren TensorFlow-Version zu benutzen. Das liegt daran, dass die jeweils aktuelle TensorFlow-Version mit der damals aktuellen jetzt aber älteren Python-Version entwickelt wurde.

Diese Aufgaben wurden mit Python 3.8.10 (64 Bit) und TensorFlow 2.6.0 getestet.



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bilderkennung

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



A2 Stundenübersicht

1. Doppelstunde:

- Hinführung zum Thema
- Grundlagen der Bilderkennung mit NN und TensorFlow
- Programmierübungen
- weiterführend: eigenständiges Projekt

1. Doppelstunde

Zeit	Phase	Beschreibung	Methode/Medien/ Anmerkung
5 Min.	Einstieg	Impuls zur Bedeutung und Problematik von Bilderkennung: Problematisierung der Schwierigkeit Bilderkennung mit Hilfe klassischer Algorithmen (<code>if... then...else</code>) zu programmieren.	Präsentation: z. B. Vielfalt der Blumenbilder
10 Min.	Vortrag / LSG / Erarbeitung 1	Lehrervortrag zu Grundlagen von Bilderkennung Möglicher begleitender Beobachtungsauftrag zur Präsentation: Beschreibe welche Vorteile die beiden vorgestellten Schichten bieten.	Präsentation: <i>convolutional backbone</i>
10 Min.	LSG	Besprechung Quellcode	
20 Min.	Erarbeitung 2	Aufgaben 1. - 3.	
10 Min.	Sicherung	Diskussion der Ergebnisse	
35	Erarbeitung 3	Aufgabe 4.	Durch Serienaufnahmen



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bilderkennung

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN





Zeit	Phase	Beschreibung	Methode/Medien/ Anmerkung
Min. / Transfer		Planung / Erstellung eines eigenen Projektes mit Anwendung/ Veränderung des bestehenden Modells	lassen sich sehr schnell viele Bilder erzeugen. Neben Gesichtserkennung bietet sich als Projekt auch die Unterscheidung von Gegenständen an.
5 Min.	Reflexion	Projektvorstellung und Reflexion	

A3 Themeneinstieg und theoretische Grundlagen

Der Themeneinstieg erfolgt über eine kurze Diskussion zur ersten Seite der unterrichtsbegleitenden Präsentation. Dabei sollte klar werden, dass bisher gelernte Programmieretechniken mit `if..else` und Schleifen Grenzen haben und nicht zum Ziel einer umfassenden Bilderkennung führen können.

Theoretische Grundlagen finden sich in der Lernaufgabe „KI programmieren im Informatikunterricht Teil I: Einführung“. Für weitere Hinweise empfiehlt sich die Lektüre der Quellen wie unter: „D3 Quellen / Lektüreliste zum Weiterlesen“ aufgeführt.



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.



B Lernaufgabe

B Lernaufgabe



B1 Unterrichtsbegleitende Präsentation



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung

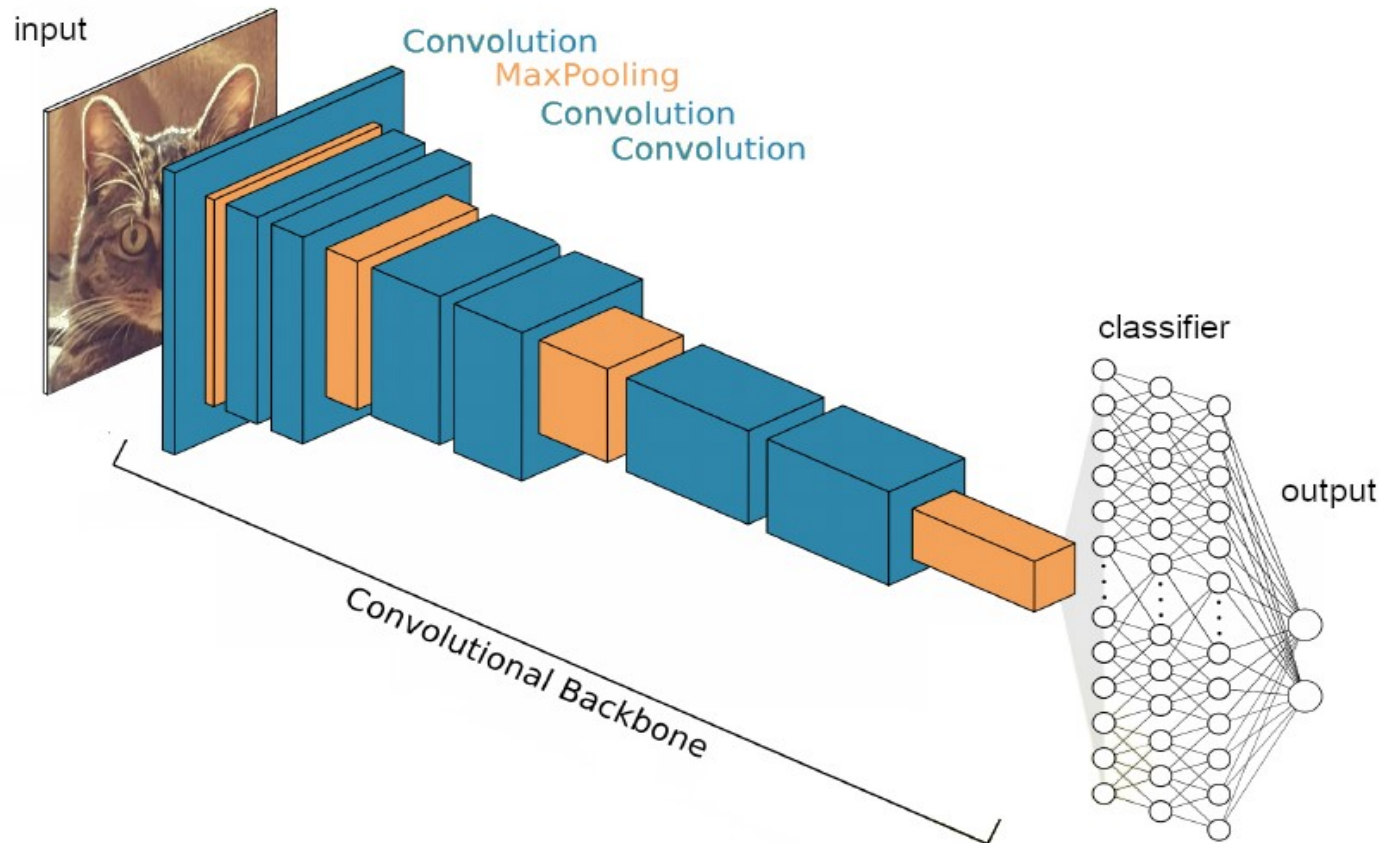


Wie programmiert man eigentlich eine Bildererkennung?



Abb.[B4]: „InOutGänseblümchen“, A. Schindler Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Unterricht" unter Verwendung weiterer Quellen (s. Bildnachweis)

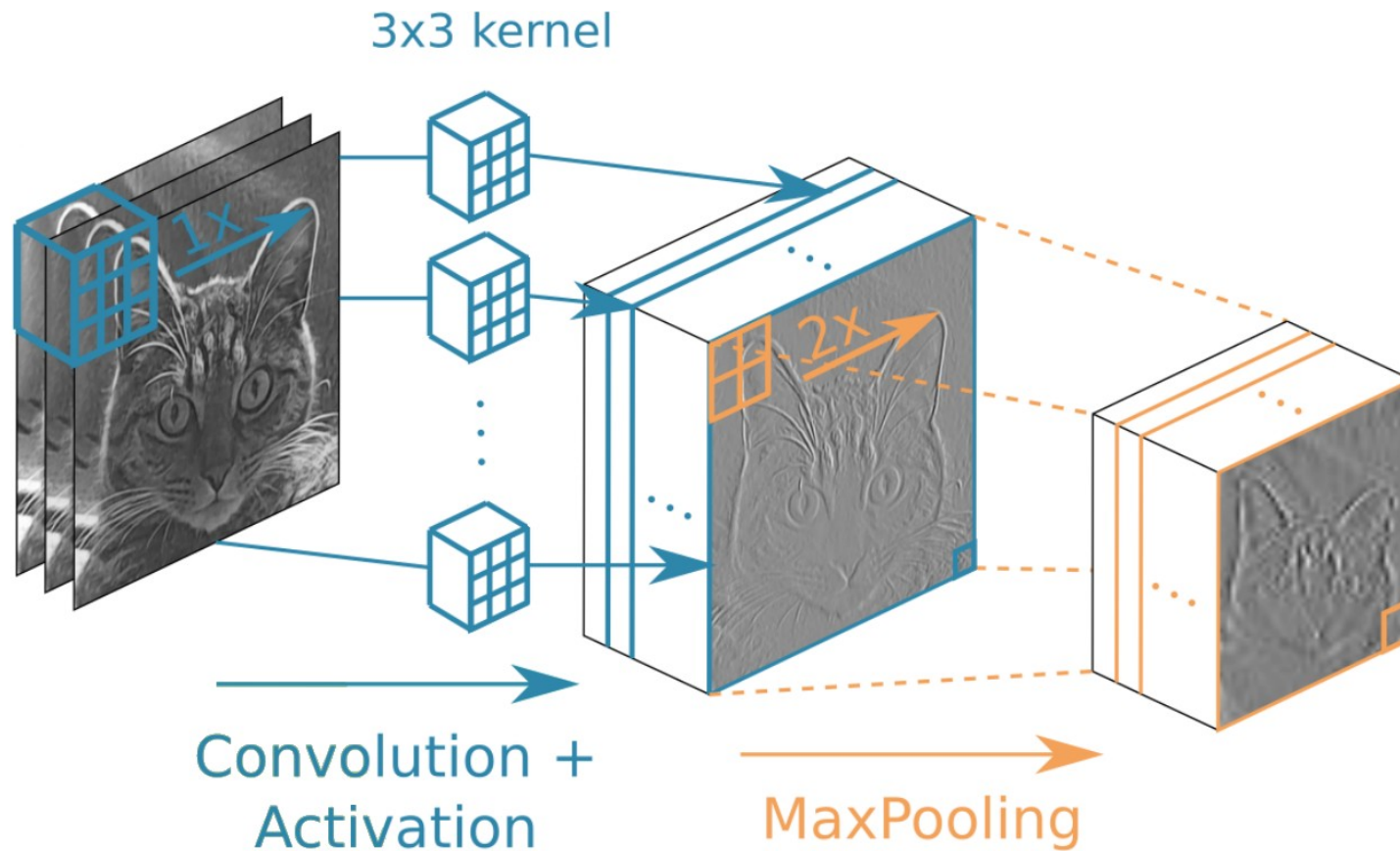
1. Bilderkennung



[B1]

Abb. [B1]: „Figure 2“ HOESER Thorsten, KUENZER, Claudia: [Object Detection and Image Segmentation with Deep Learning on Earth Observation Data](#), [CC BY 4.0](#), verändert v. A. Schindler (Ausschnitt)

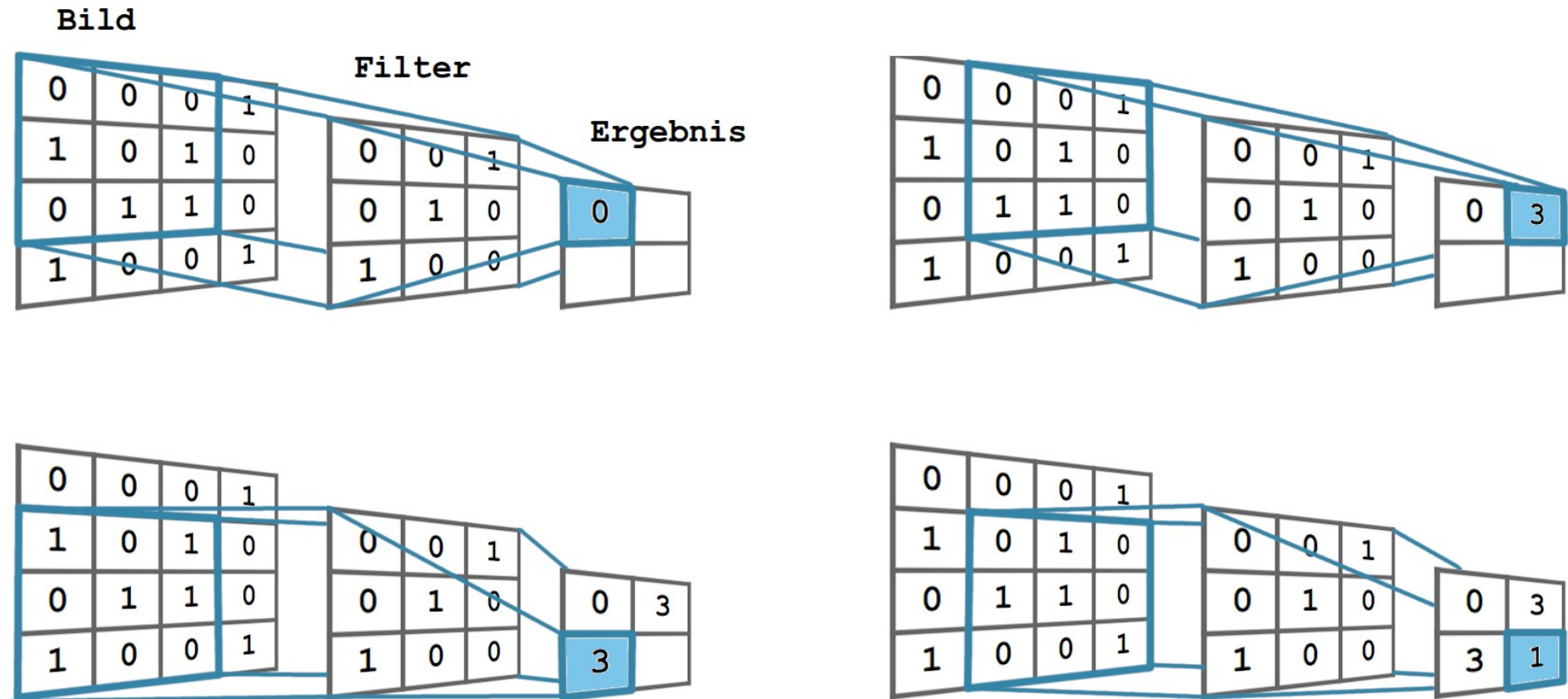
1. Convolutional Backbone



[B1]

Abb. [B1]: „Figure 2“ Thorsten, KUENZER, Claudia: [Object Detection and Image Segmentation with Deep Learning on Earth Observation Data](#), CC BY 4.0, verändert v. A. Schindler (Ausschnitt)

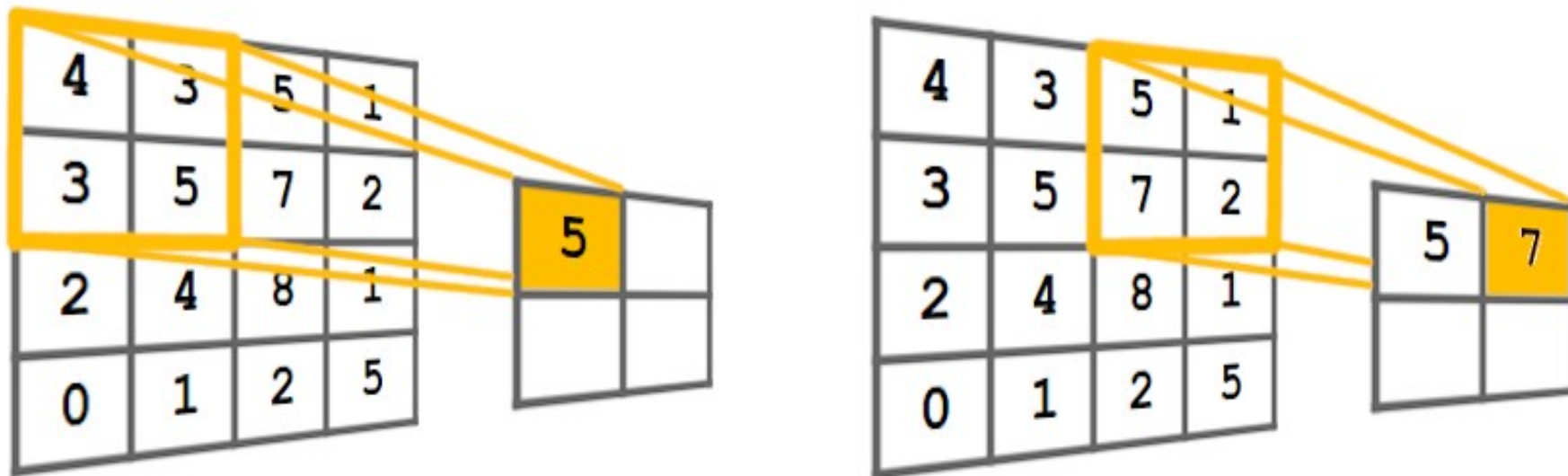
1. Data Processing: Convolution



[B3]

Abb. [B3]: „ConvolutionInBluePic4x4Filter3x3“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Unterricht II"

1. Data Processing: Max Pooling



[B2]

Abb. [B2]: „MaxPoolingInOrangePic4x4Pool2x2“, Alexander Schindler, Lizenz [CC BY-SA4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Unterricht II"

2. All hands on Code

```
1  # Bilder klassifizieren - last edited: 15.11.2021, version: 1.2, author: A. Schindler #
2
3  from PIL import Image
4  import tensorflow as tf
5  import numpy as np
6
7  ### Daten vorbereiten ###
8  trainBilder = tf.keras.preprocessing.image_dataset_from_directory('data/train',
9      image_size=(200,200))
10 anzahlKlassen = len(trainBilder.class_names)
11 print("Folgende", anzahlKlassen, "Klassen wurden erkannt:", trainBilder.class_names)
12
13 ### Model erstellen ###
14 model = tf.keras.Sequential()
15 model.add(tf.keras.layers.experimental.preprocessing.Rescaling(1./255))
16 model.add(tf.keras.layers.Conv2D(16, 3, padding='same', activation='relu'))
17 model.add(tf.keras.layers.MaxPooling2D())
18 model.add(tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu'))
19 model.add(tf.keras.layers.MaxPooling2D())
20 model.add(tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'))
21 model.add(tf.keras.layers.MaxPooling2D())
22 model.add(tf.keras.layers.Flatten())
23 model.add(tf.keras.layers.Dense(128, activation='relu'))
24 model.add(tf.keras.layers.Dense(anzahlKlassen))
```

2. All hands on Code

```
25 model.compile(optimizer='adam', loss = tf.keras.losses.SparseCategoricalCrossentropy
26                 (from_logits=True), metrics=['accuracy'])
27
28 ### Model trainieren ###
29 model.fit(trainBilder, epochs=10)
30
31 ### Model testen ###
32 img = tf.keras.preprocessing.image.load_img('data/imageTestDand.jpg', target_size=(200, 200))
33 img_array = tf.keras.preprocessing.image.img_to_array(img)
34 img_array = np.array([img_array])
35 predictions = model.predict(img_array)
36 score = tf.nn.softmax(predictions[0])
37
38 # Ausgabe
39 Image.open('data/imageTestDand.jpg').show()
40 print("Mit einer Wahrscheinlichkeit von {:.2f} % ist das Bild ein: {}".format(100 *
41 np.max(score), trainBilder.class_names[np.argmax(score)]))
```

3. Aufgaben

Aufgaben:

1. Beschreibe den Aufbau des `models`. Was gehört zum NN was zum *convolutional backbone*?
2. Führe den Code aus.
3. Verändere das *convolutional backbone* und NN. Beschreibe die Auswirkungen. Versuche möglichst hohe Erkennungsraten zu erreichen.
4. Erstelle ein eigenes Projekt z. B. zur Gesichtserkennung. Erstelle dir dazu eigene Daten.

4. Diskussion

Bildererkennung (*computer vision*) kann dazu benutzt werden um Menschen das Leben zu retten (Krankheitserkennung, autonomes Fahren usw.).

Bildererkennung kann aber auch benutzt werden um Einzelne oder ganze Gesellschaften total zu überwachen oder sogar um Menschen zu töten (Militär).

Diskutiere in diesem Zusammenhang die Verantwortung der Softwareentwickler: innen.

5. Sei ein weißer Roboter.

Gesichtserkennung Tipp für Berufseinsteiger: Sei ein weißer Roboter, Ein Netzwelt Newsletter von Patrick Beuth, 15.11.2021

Quelle: www.spiegel.de/netzwelt/web/gesichtserkennung-tipp-fuer-berufseinsteiger-sei-ein-weisser-roboter-a-112f0ccf-5ac9-49d4-97ce-11f0f1c89068

Juristen in den USA, die Dokumente für die Verwendung in Prozessen vorbereiten, haben das vor der Pandemie häufig in speziellen Büros mit Zugangskontrollen gemacht. In Pandemiezeiten sollte diese Arbeit im Homeoffice erledigt werden. Da es um Unternehmensinterna, E-Mails und andere sensible Dokumente geht, die nicht jeder sehen darf, wurden die Juristen dabei von einer Software überwacht.

Bei Camille Anidi, einer schwarzen Anwältin aus Long Island im Bundesstaat New York passierte Folgendes: schaute sie zu lange nach links, wurde sie aus dem System geworfen und musste sich mit einem erneuten Scan ihres Gesichts von drei Seiten wieder einloggen. Kam ihr Hund ins Zimmer, passierte das gleiche. Trug sie ihre Haare nicht offen, hielt dies die Software für verbotene Aufnahmegeräte und loggte sie ebenfalls aus. Dies geschah bis zu 25 Mal täglich.



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

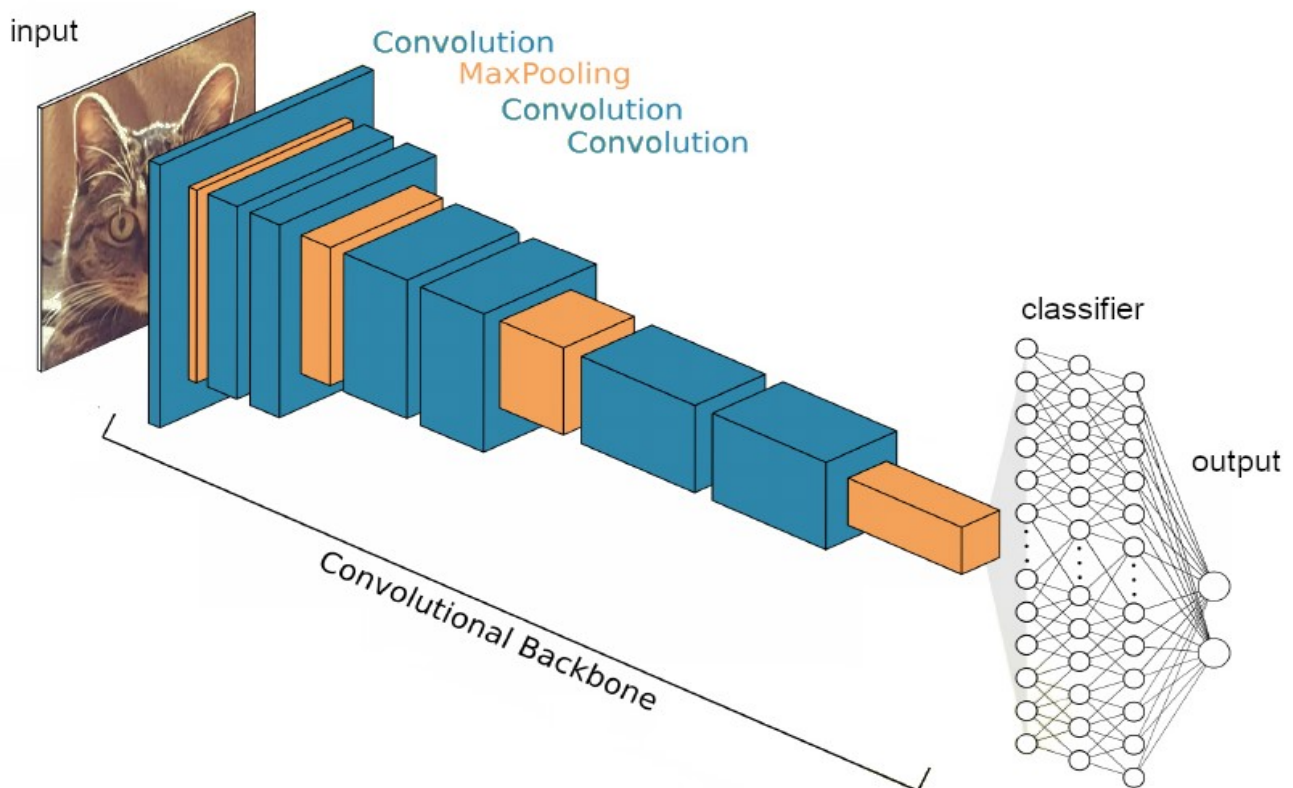
iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung

Senatsverwaltung
für Bildung, Jugend
und Familie:

BERLIN



Computer Vision – Bilderkennung mit Deep learning



[B1]

Abb. 1: Aufbau eines *convolutional backbone*s mit neuronalem Netzwerk NN

Aufgaben:

1. Beschreibe den Aufbau des `models`. Was gehört zum NN was zum *convolutional backbone*.
2. Führe den Code aus.
3. Verändere das *convolutional backbone* und das NN, beobachte die Auswirkungen. Versuche möglichst hohe Erkennungsraten zu erreichen.
4. Erstelle ein eigenes Projekt z. B. zur Gesichtserkennung. Erstelle dir dazu eigene Daten.

Abb. 1: [B1] „Figure 2“ HOESER Thorsten, KUENZER, Claudia [2020]: [Object Detection and Image Segmentation with Deep Learning on Earth Observation Data](#), CC BY 4.0, verändert v. A. Schindler (Ausschnitt)



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bilderkennung

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



Gesichtserkennung Tipp für Berufseinsteiger: Sei ein weißer Roboter,

Ein Netzwelt Newsletter von Patrick Beuth, 15.11.2021

Juristen in den USA, die Dokumente für die Verwendung in Prozessen vorbereiten, haben das vor der Pandemie häufig in speziellen Büros mit Zugangskontrollen gemacht. In Pandemiezeiten sollte diese Arbeit im Homeoffice erledigt werden. Da es um Unternehmensinterna, E-Mails und andere sensible Dokumente geht, die nicht jeder sehen darf, wurden die Juristen dabei von einer Software überwacht.

Bei Camille Anidi, einer schwarzen Anwältin aus Long Island im Bundesstaat New York passierte Folgendes: schaute sie zu lange nach links, wurde sie aus dem System geworfen und musste sich mit einem erneuten Scan ihres Gesichts von drei Seiten wieder einloggen. Kam ihr Hund ins Zimmer, passierte das gleiche. Trug sie ihre Haare nicht offen, hielt dies die Software für verbotene Aufnahmegeräte und loggte sie ebenfalls aus. Dies geschah bis zu 25 Mal täglich.

Quelle: www.spiegel.de/netzwelt/web/gesichtserkennung-tipp-fuer-berufseinsteiger-sei-ein-weisser-roboter-a-112f0ccf-5ac9-49d4-97ce-11f0f1c89068

5. Bewerte den hier beschriebenen Einsatz der Gesichtserkennung.



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung



C Bezug zum Rahmenlehrplan

C1 Didaktischer Kommentar

Ziel der Aufgabe ist es, den Schülern einen Einstieg in die Programmierung von Neuronalen Netzwerken (NN) zur Bilderkennung zu ermöglichen.

Der Code ist so einfach wie möglich gehalten, damit der Einstieg in die Programmierung mit TensorFlow leicht fällt und auch im Informatikunterricht der Sek I möglich ist. Der Code bietet Anknüpfungspunkte zum Arbeiten mit Schleifen, if-Abfragen und Listen und lässt sich somit auch für den Anfängerunterricht einsetzen. Einige Vorerfahrungen mit NN, Python und Softwareentwicklung sollten aber vorhanden sein, somit ist die Aufgabe für Schüler der Sek II besser geeignet. Die Aufgabe ist skalierbar, d. h. sie kann um eigene Ideen (z. B. Datensätze, Problemstellungen) erweitert werden und sie kann und soll somit Basis für eigene Projekte sein. Leistungsstarke SuS können entsprechend komplexe Projekte mit eigenen NN umsetzen. Leistungsschwächere SuS übernehmen das bestehende NN und die existierende Daten.

Nicht zuletzt ist mit der Thematisierung von Bilderkennung durch KI ein Ausblick auf die Auswirkungen von KI auf die eigene Lebenswelt der Schüler möglich. Somit ist die Aufgabe auch im Bereich Informatik und Gesellschaft zu verorten.

Lernvoraussetzungen	<ul style="list-style-type: none">• Die SuS können einfache Algorithmen in Python programmieren.• oder: besitzen vertiefte Programmierkenntnisse in einer anderen Programmiersprache• Die SuS können mit Variablen umgehen.• Die SuS haben eine grundlegende Vorstellung von Algorithmen und Datenstrukturen.• Die SuS können eine IDE benutzen.• Die SuS haben ein grundsätzliches Verständnis für ML wie in der Lernaufgabe: „KI programmieren im Informatikunterricht Teil 1: Einführung“
---------------------	---

C2 Bezüge zum Rahmenlehrplan Informatik

Kompetenzen	Standards (Die Schülerinnen und Schüler können...)
Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen</p> <p>Kompetenz: Historische und aktuelle Entwicklungen der Informatik beurteilen</p> <p>Standard H: aktuelle Entwicklungen bewerten, aus ihnen mögliche Trends ableiten und Auswirkungen in der Zukunft beschreiben die Chancen und Risiken der modernen Entwicklungen für eine demokratische Gesellschaft bewerten</p>
Mit Informationen umgehen	
Informatiksysteme verstehen	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Informatiksysteme verstehen</p> <p>Kompetenz: Technische Grundlagen erläutern und anwenden</p> <p>Standard G/H: ein einfaches Informatiksystem entwerfen, modifizieren bzw. realisieren, z. B.: Verkehrsampelmodell, Robotermodelle</p>
Informatisches Modellieren	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Informatisches Modellieren</p> <p>Kompetenz: Informatische Modelle analysieren und bilden</p> <p>Standard F: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen</p> <p>Standard H: beurteilen, ob das selbst erstellte Modell problemadäquat ist</p>
Problemlösen	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Problemlösen</p> <p>Kompetenz: Programme entwerfen und realisieren</p> <p>Standard F: eine Programmierumgebung verwenden</p>

	<p>Standard H: Algorithmen entwerfen, implementieren und beurteilen</p>
Kommunizieren und kooperieren	<p>Bezug zum RLP Sek I: Kompetenzbereich: Kommunizieren und Kooperieren – Teamarbeit organisieren und koordinieren Kompetenz: Arbeitsergebnisse dokumentieren und präsentieren Standard G: adressatengerecht mit Softwareunterstützung präsentieren</p>

Unterrichtsfach	Informatik
Jahrgangsstufe/n	Sek I: 10 Sek II: IN-3
Niveaustufe/n	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Informatisches Modellieren Kompetenz: Informatische Modelle analysieren und bilden Standard F: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen Standard G: ein Modell selbst erstellen Standard H: beurteilen, ob das selbst erstellte Modell problemadäquat ist</p> <p>Bezug zum RLP Sek II: 3. Kurshalbjahr (IN-3) Grundlagen der Informatik und Vertiefungsgebiet: V5 Künstliche Intelligenz</p>
Zeitraumen	Eine Doppelstunde.
Thema	Bildererkennung (<i>computer vision</i>) mit Neuronalen Netzen.

Kontext	<ul style="list-style-type: none"> • Beruf und Arbeitswelt • Überwachung • Robotik
---------	---



CC BY SA 4.0
 Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.



	<ul style="list-style-type: none"> • autonome Agenten • Gesellschaft
Schlagwörter	Programmierung, Informatik und Gesellschaft, Informatiksystem verstehen, Problemlösen, mit Informationen umgehen, Modellbildung, künstliche Intelligenz, KI, Deep Learning, DL, TensorFlow, Keras, Python, Daten

C3 Bezüge zum Basiscurriculum Medienbildung¹

Standards des BC Medienbildung	Die Schülerinnen und Schüler können ...
Präsentieren	<ul style="list-style-type: none"> • Arbeitsergebnisse vorstellen • Reflektierter Technologieeinsatz
Reflektieren	<ul style="list-style-type: none"> • Chancen und Risiken von Geschäftsaktivitäten im Internet untersuchen und Schlussfolgerungen für eigene Geschäftsaktivitäten ziehen. • Die Möglichkeiten und Methoden medialer Manipulation exemplarisch analysieren. • An aktuellen und historischen Beispielen den ökonomischen und politischen Einfluss von Medien(-konzernen) auf Meinungsbildungsprozesse in der Gesellschaft exemplarisch analysieren.

C4 Bezüge zu übergreifenden Themen²

Berufs- und Studienorientierung	Data Engineer, Data Analyst, Data Scientist, Machine Learning Engineer, Softwareentwickler
Verbraucherbildung	Präsenz von neuronalen Netzen in alltäglichen Geräten, Anwendungen

1 vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 15-22, Berlin, Potsdam 2015

2 vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 24ff, Berlin, Potsdam 2015



CC BY SA 4.0
 Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.





C5 Bezüge zu anderen Fächern:

- Gesellschaftswissenschaften: Einsatz von NN in Medizin, Interaktion von NN und Menschen, Auswirkung von Überwachung und Gesichtserkennung, Philosophie und Ethik bei der Bewertung möglicher Auswirkungen von Gesichtserkennung
- Bildererkennung in der Medizin z. b. Krebserkennung bei Mammographie



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.



D Anhang

D1 „Material“ für den Einsatz dieser Lernaufgabe

- Für SuS eine entsprechend leistungsfähige Python IDE, es empfiehlt sich Spyder.
- Python (derzeit Januar 2022 ab Version 3.8, 64 Bit) mit installierter TensorFlow (ab 2.6) Bibliothek
- Da die jeweils aktuelle TensorFlow Bibliothek in einer nicht aktuellen Pythonversion entwickelt wird, hilft es bei (unklaren) Problemen beim compilieren eine eher ältere Pythonversion zu verwenden.
- Zeile **8 / 39**: Dateipfade hängen vom Betriebssystem und dem tatsächlich genutzten Verzeichnis der IDE ab. Dies variiert in abhängigkeit von der IDE, die ja z.T. auch unterschiedlich konfiguriert sein können. Beispielpfad unter Windows
`Pfad_zum_Bilderordner\\flower_photos`
- Zeile **39**: `imageTestDand.jpg` ist ein beliebiges Bild eines Löwenzahns im Verzeichnis `data`.

D2 Hinweise / Musterlösung der Lernaufgabe

1. Beschreibe den Aufbau des `models`. Was gehört zum NN, was zum *convolutional backbone*?
 - Zeile **15**: Normalisierung, alle Werte werden in den Bereich von 0-1 umgerechnet
 - Zeile **16 – 21**: *convolutional backbone*
 - Zeile **22**: mehrdimensionale Datenstruktur wird in eine eindimensionale Datenstruktur umgeformt.

NN:

- Zeile **23**: hidden Layer mit 128 Neuronen
- Zeile **22**: output layer Anzahl der Neuronen entspricht Anzahl der Klassen

2. Führe den Code aus.

- Die Ausführung lässt sich Beschleunigen durch:
 - Verkleinerung der Bildgröße Zeile 9 und 32 (im Original 200 x 200)
 - Reduzierung der Bilder bzw. Bilderklassen
 - Reduzierung der Epochenanzahl
- Erklärung der Ausgabe (Auszug):

```
Found 3670 files belonging to 5 classes.
```



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung



```
Folgende 5 Klassen wurden erkannt: ['daisy',
'dandelion', 'roses', 'sunflowers', 'tulips']
Epoch 1/10
115/115 [=====] - 19s 160ms/step -
loss: 1.2365 - accuracy: 0.4728
Epoch 2/10
115/115 [=====] - 18s 154ms/step -
loss: 0.9247 - accuracy: 0.6406
Epoch 3/10
 91/115 [=====>.....] - ETA: 3s - loss:
0.8062 - accuracy: 0.6899
```

- **115** = 3670 (Anzahl der Bilder) / 32 (Batchsize)
 - **115:** : Größe des Trainingsets
 - **ETA** : Estimated Time of Arrival.
 - **loss** : Unterschied zwischen der Vorhersage (`predict`) und dem richtigen *label*.
 - **accuracy** : Metrik die zur Berechnung der Häufigkeit verwendet werden, mit der die Vorhersage mit den richtigen *labeln* übereinstimmt.
3. Verändere convolutional backbone und NN. Beobachte die Auswirkungen. Versuche möglichst hohe Erkennungsraten zu erreichen.
- Hier muss man einfach Erfahrungen sammeln und verschiedene Parameter verändern, Schichten weglassen oder hinzufügen, z. B. weitere *hidden layer*.
4. Erstelle ein eigenes Projekt z. B. zur Gesichtserkennung. Erstelle dir dazu eigene Daten. Diese Aufgabe kann auch mehr Zeit in Anspruch nehmen und über mehrere Unterrichtsstunden ausgedehnt werden. Mögliche Bilderquellen:
- a) Serienaufnahmen mit Handys können in kurzer Zeit viele Bilder generieren. z. B. von Gegenständen oder Gesichtern.
 - b) Weitere Datenquellen:
 1. www.tensorflow.org/datasets/catalog/overview
 2. <https://knowyourdata-tfds.withgoogle.com/>
 3. www.image-net.org
5. Bewerte den hier beschriebenen Einsatz der Gesichtserkennung. In dieser Bewertung können zahlreiche Aspekte einfließen z. B.:
- Menschen mit heller Hautfarbe werden von Kameras i. d. R. besser aufgenommen. Die Vermutung liegt nahe, dass bei der Entwicklung der

Kameras beim Testen v. a. Menschen mit heller Hautfarbe aufgenommen werden, so dass sie ungewollt dahingehend optimiert sind. Das gleiche gilt für das Training des zugrundeliegenden NN. Hier findet durch das NN ungewollt eine **Diskriminierung** statt.

- **Arbeitnehmerrechte** stehen ihr dem Schutzbedürfnis von Daten entgegen.
- **Totale Überwachung** von Arbeitnehmern kann jetzt schon dystopische Ausmaße annehmen, wie man in dem Text „Sei ein weißer Roboter“ nachvollziehen kann.

Zusatzaufgaben

6. Beschreibe Dein `model`. Benutze dazu den Befehl: `model.summary()`.

Die Ausgabe von `model.summary()` zeigt den Aufbau des models. Die verschiedenen Layer und ihre Parameter.

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 200, 200, 3)	0
conv2d (Conv2D)	(None, 200, 200, 16)	448
max_pooling2d (MaxPooling2D)	(None, 100, 100, 16)	0
conv2d_1 (Conv2D)	(None, 100, 100, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 50, 50, 32)	0
conv2d_2 (Conv2D)	(None, 50, 50, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 25, 25, 64)	0
flatten (Flatten)	(None, 40000)	0
dense (Dense)	(None, 128)	5120128
dense_1 (Dense)	(None, 5)	645
Total params: 5,144,357		
Trainable params: 5,144,357		
Non-trainable params: 0		

7. Verändere die Anzahl der Layer / Knoten. Beschreibe die Auswirkungen auf das Ergebnis und erkläre die Veränderungen.



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung



- in einem explorativen Einsatz kann die Veränderung des Modells geübt werden. Veränderungen können in Abhängigkeit von den Daten unterschiedliche Auswirkungen haben.
8. Ermittle wo sinnvolle Unter- / Obergrenze für die Knotenanzahl liegt.
 - Zu wenige Knoten verhindern die Erkennung, da das NN nicht genug Gewichte zum lernen hat.
 - Zu viele Knoten können zu *overfitting* und Performanceverlusten führen.
 9. Überlege, welche Auswirkungen eine zu große Knotenanzahl haben kann.
 - Es besteht die Gefahr, dass das NN die Bilder auswendig lernt (*overfitting*) und somit im Training sehr gute beim Test aber sehr schlechte Ergebnisse liefert.
 10. Wechsel die Optimizer aus. Beobachte mögliche Veränderungen.
 - Das Wechseln der Optimizer kann bessere Ergebnisse liefern. Welchen Optimizer man auswählt beruht auf Erfahrung und auf Versuch und Irrtum.
 - Der Wechsel der Optimizer wird in diesem einfachen Beispiel keine größeren Auswirkungen haben, man kann aber den grundsätzlichen Einsatz darstellen.
 - Es empfiehlt sich hier den Aufbau der TensorFlow API www.tensorflow.org/api_docs/python/tf/all_symbols vorzustellen.
 - Weitere Optimizer finden sich hier: www.tensorflow.org/api_docs/python/tf/keras/optimizers

D3 Quellen / Lektüreliste zum Weiterlesen

- www.tensorflow.org/tutorials/images/classification
- MORONEY, Laurence[2020]: AI and Machine Learning for Coders, A Programmer's Guide to Artificial Intelligence.
- 3700 Blumenbilder:
http://download.tensorflow.org/example_images/flower_photos.tgz

Bildtitel	Seite	Quelle
[B0]	1	„E-V-A“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Unterricht"
[B1]	8, 18	„Figure 2“ verändert nach: HOESER, Thorsten, KUENZER, Claudia [2020]: Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends, in: remote sensing, Figure 2, https://doi.org/10.3390/rs12101667 © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution CC BY 4.0
[B2]	11	„MaxPoolingInOrangePic4x4Pool2x2“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Unterricht II"
[B3]	10	„ConvolutionInBluePic4x4Filter3x3“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Unterricht II"
[B4]	7	„InOutGänseblümchen“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Unterricht" unter Verwendung weiterer Quellen [B5], [B6], [B7], [B8], [B9], [B10] s. Bildnachweis
[B5]	7	"daisys" by osde8info, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/576d8df4-4328-4618-82a4-0e38b1710739 https://www.flickr.com/photos/8764442@N07/795087477
[B6]	7	"Daisies and shed" by cdanna2003, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/d24b3ff1-82e7-4341-8b1a-e0ea85725723 https://www.flickr.com/photos/49565454@N00/835750256



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bildererkennung

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



Bildtitel	Seite	Quelle
[B7]	7	"daisy" by dasWebweib, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/8f1ee061-e603-4b9f-8af1-14e781a6c777 https://www.flickr.com/photos/21059112@N02/16593982857
[B8]	7	"Common daisy" by hedera.baltica, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/a9149f7e-2347-41a7-81ca-f2da7d296db9 https://www.flickr.com/photos/125741467@N05/30755571987
[B9]	7	"our lady of the daisy" by jasleen_kaur, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/363319c9-2adc-4ff2-8245-14d78e6f0e57 https://www.flickr.com/photos/25384802@N08/6884975451
[B10]	7	„daisy wallpaper" by .robbie, Lizenz CC BY-SA 2.0 . abgerufen: 11.3.2022 https://search.openverse.engineering/image/2fe2317e-f8d6-46ae-939d-23741b308f90 https://www.flickr.com/photos/14821912@N00/2365428551



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie Fachset Informatik
KI Programmieren im Unterricht Teil 2:
Bilderkennung

Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN

