Was sieht die KI? (Objekterkennung)

Kapitel online

Lade ein Foto hoch oder starte die Kamera: COCO-SSD zeichnet live beschriftete Rahmen um alles, was es erkennt — und zeigt dir die rohen Zahlen dahinter. Bild und Kamerabild bleiben dabei auf deinem Gerät.

Interaktiv · Kapitel 45 von 50 · ca. 15 Min. · 300 XP

Einfach gesagt: Du zeigst der KI ein Foto oder deine Kamera, und sie malt Rahmen um alles, was sie erkennt — mit einer Prozentzahl, wie sicher sie sich ist.

Ein Auto bremst, weil vor ihm jemand auf die Straße tritt. Du tippst „Hund“ in die Suche deiner Handy-Galerie und bekommst 240 Fotos, die nie jemand beschriftet hat. Die Selbstbedienungskasse erkennt deine Avocado — ganz ohne Barcode. Drei Alltagssituationen, ein und dieselbe Technik dahinter: **Objekterkennung**.

Und das Verrückte: Genau diese Fähigkeit holst du dir in den nächsten Minuten in *deinen* Browser. Kein Cloud-Dienst, kein Konto, keine Rechnung — ein 19 MB großes Modell namens **COCO-SSD**, das seit 2017 auf 80 Alltagsdinge trainiert ist, lädt gleich in dieses Tab und schaut sich dein Foto an.

Roter Faden: Von der Anwendung, die jeder kennt, runter bis zum Tensor — und wieder hoch. Am Ende siehst du dein eigenes Foto als das, was es für die KI ist: ein Stapel Zahlen. Und ihre Antwort ebenfalls: zwei Zahlenlisten.

Was du hier lernst

- Der Unterschied zwischen **Klassifikation** („was ist drauf?“) und **Detektion** („was ist *wo*?“)
- Was ein Erkennungsmodell **wirklich** ausspuckt — Rahmen, Punktzahl, Klassennummer. Sonst nichts.
- Was der Prozentwert bedeutet — und was er **nicht** bedeutet
- Warum dein Bürostuhl erkannt wird, dein 3D-Drucker aber nicht
- Im Mitmach-Teil: eigenes Foto oder Live-Kamera mit beschrifteten Rahmen — lokal in deinem Browser

Für den Unterricht

Einstieg (3 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (9 Min): die interaktiven Module auf herrki.de/430_objekterkennung ausprobieren.

Sichern mit Quiz (3 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Klassifikation vs. Detektion — der Unterschied in drei Sätzen

Ein **Klassifikator** beantwortet genau eine Frage: „Was ist auf diesem Bild?“ — eine Antwort fürs ganze Bild, wie das Ziffern-Netz aus Kapitel 420, das entscheidet: das ist eine 7. Ein **Detektor** beantwortet drei Fragen auf einmal: **was** ist zu sehen, **wo** genau steckt es im Bild, und **wie sicher** ist sich das Netz dabei. Deshalb bekommst du gleich nicht ein Wort, sondern für jeden Fund einen Rahmen, ein Etikett und eine Prozentzahl.

	Klassifikation	Detektion (heute)
Frage	Was ist drauf?	Was ist *wo* — und wie sicher?
Antwort	1 Wort pro Bild	0 bis 20 Funde pro Bild
Rohausgabe	Liste von Wahrscheinlichkeiten	Rahmen-Koordinaten + Punktzahlen + Klassennummern
Typisch für	Ziffern lesen, Katze/Hund	Assistenzsysteme, Kassen, Foto-Suche

Technisch ist ein Detektor ein Klassifikator mit einem zweiten Kopf: Derselbe Faltungs-Turm (die „Filter“ aus Kapitel 420) schaut sich das Bild an — aber am Ende hängen zwei Ausgänge dran. Einer rät die Klasse, der andere die vier Zahlen des Rahmens. „SSD“ heißt *Single Shot Detector*: ein einziger Durchlauf durchs Netz, alle Funde auf einmal.

Dein Werkzeug: COCO-SSD im eigenen Browser

Am Gerät ausprobieren: herrki.de/430_objekterkennung#abschnitt-3

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Runter zum Tensor: Was liefert das Netz WIRKLICH?

Die bunten Rahmen oben sind Deko — die haben *wir* gemalt. Was das Netz zurückgibt, ist deutlich nüchterner. Erst mal die Reise deines Bildes **hinein**:

1. Dein Foto
z. B. 1200 × 900 Pixel

→

2. Tensor
[1, 900, 1200, 3]
ganze Zahlen 0–255

→

3. Faltungs-Turm
MobileNet-V2, Millionen Multiplikationen

→

4. Zwei Zahlenlisten
Rahmen + Punktzahlen

Schritt 2 ist exakt das, was du in „Bilder als Tensoren“ gebaut hast: Höhe × Breite × 3 Farbkanäle, jeder Wert eine Zahl von 0 bis 255. Die vorderste `1` ist die Stapelgröße („ein Bild bitte“). Für das Netz gibt es kein Foto — es gibt nur diesen Zahlenblock.

Und heraus kommt pro Fund genau das hier:

```
// Beispiel – nach der ersten Erkennung steht hier DEIN oberster Fund
{
  bbox: [ 142, 88, 310, 255 ], // x, y, breite, hoehe – in Pixeln des Originalbildes
  score: 0.87, // Punktzahl 0..1 – wie stark das Netz auf diese Klasse tippt
  class: 'dog' // eine von 80 Klassennummern, hier als Name
}
```

Mehr ist da nicht. `bbox` sind vier Pixelzahlen: linke obere Ecke, dann Breite und Höhe des Kastens. `score` ist eine einzige Fließkommazahl. Und `class` ist im Rohzustand nicht einmal ein Wort, sondern die **Nummer 17** — dass daraus „Hund“ wird, macht eine schlichte Übersetzungstabelle, die in diesem Kapitel im JavaScript liegt (80 Zeilen, eine pro Klasse, von uns auf Deutsch übersetzt).

Kernsatz: Auch „Sehen“ ist am Ende Tensor-Rechnung. Zahlen gehen rein, Zahlen kommen raus — ein paar Koordinaten und ein paar Punktzahlen. Die *Bedeutung* („da steht ein Mensch, pass auf!“) entsteht erst in dem Programm, das diese Zahlen benutzt. Und in deinem Kopf.

Ehrlich bleiben: was dieses Modell NICHT kann

- **Es kennt genau 80 Dinge.** Der COCO-Datensatz wurde 2014–2017 zusammengestellt, die Liste ist seither eingefroren. Kein Gabelstapler, kein Rollator, kein Bildschirm-am-Monitorarm, kein E-Scooter, kein 3D-Drucker. Was nicht im Katalog steht, kann nicht erkannt werden — im besten Fall schweigt das Netz, im schlechteren macht es daraus das ähnlichste bekannte Ding.
- **Ungewohnte Blickwinkel und schlechtes Licht brechen ihm das Genick.** Trainiert wurde an Fotos aus Menschenperspektive. Von oben, im Gegenlicht, halb verdeckt oder unscharf sinken die Punktzahlen rapide.
- **Kleine und viele Objekte sind schwer.** Das schlanke Handy-Modell rastert das Bild grob ab; ein Vogel weit hinten am Himmel fällt schlicht durchs Raster.
- **Und das Wichtigste: Das ist kein Verstehen der Szene.** Das Netz weiß nicht, dass ein Messer neben einem Kuchen etwas anderes bedeutet als ein Messer in einer Faust. Es kennt keine Absicht, keine Gefahr, keinen Zusammenhang — nur Muster, die statistisch zu 80 Etiketten passen.

Kommt dir bekannt vor? Genau dieselbe Logik steckt hinter den Halluzinationen der Sprachmodelle: Ein Netz gibt *immer* eine Antwort mit Zahlen dahinter, auch wenn die Frage außerhalb seiner Erfahrung liegt. „87 %“ heißt nicht „das stimmt zu 87 %“ — es heißt „von meinen 80 Schubladen passt diese am besten“. Ein Modell, das nur 80 Schubladen hat, hat für deinen 3D-Drucker eben keine — und legt ihn trotzdem irgendwo hinein.

Der komplette Katalog — mehr sieht dieses Modell nicht:

Zurück zur Anwendung

Zurück zum Anfang: das bremsende Auto, die Galerie, die auf „Hund“ 240 nie beschriftete Fotos findet, die Selbstbedienungskasse mit deiner Avocado. In allen drei Fällen läuft genau das, was du gerade selbst gestartet hast — ein Netz liefert Rahmen, Klassennummer und Punktzahl, alles andere ist ganz normale Programmierung drumherum. Du kannst ein Erkennungsmodell jetzt selbst in eine Seite holen, seine Rohausgabe lesen und eine eigene Schwelle setzen, statt einer Prozentzahl blind zu glauben — und du weißt, warum dein 3D-Drucker durchfällt: Wer nur 80 Schubladen hat, legt alles in eine davon.

Bis hierhin hat die KI *Dinge* gefunden. Im nächsten Kapitel „Dein Körper als Tensor (Pose)“ wird aus dem groben Rahmen um eine Person etwas viel Feineres: 17 Punkte, 51 Zahlen — und diesmal bist du selbst der Tensor.

Zum Weiterlesen

- Redmon u. a. 2015, *„You Only Look Once“* (YOLO) — arxiv.org/abs/1506.02640 — die Arbeit, die aus Detektion einen einzigen Durchlauf statt tausender Bildausschnitte gemacht hat.
- Liu u. a. 2015, *„SSD: Single Shot MultiBox Detector“* — arxiv.org/abs/1512.02325 — das „SSD“ aus COCO-SSD: genau der Bauplan mit den zwei Ausgängen, den du hier benutzt hast.
- *COCO — Common Objects in Context* — cocodataset.org — der Datensatz hinter den 80 Klassen; dort steht schwarz auf weiß, was drin ist und was eben nicht.
- TensorFlow.js, Modell *coco-ssd* — github.com/tensorflow/tfjs-models (github.com/tensorflow/tfjs-models/tree/master/coco-ssd) — das Modell aus diesem Kapitel samt Schnittstelle, wenn du es in ein eigenes Projekt einbauen willst.

Kurzer Check – mit Lösung

Lösungen für Lehrkräfte – für Lernende die Online-Version nutzen.

Frage 1: Über deinem Hund steht „Hund 91 %“. Was bedeutet diese Zahl?

- In 91 von 100 Fällen ist dort tatsächlich ein Hund — eine geprüfte Trefferquote.
- **✓ Es ist die Punktzahl (Konfidenz), mit der das Netz auf diese Klasse tippt — eine Selbsteinschätzung, keine Garantie.**
- 91 % des Rahmens sind mit dem Hund gefüllt.

Der Score ist die eigene Einschätzung des Netzes, wie gut das Muster zu einer seiner 80 Klassen passt — errechnet aus dem Bild, nicht aus der Wirklichkeit. Ein Modell kann sich mit 95 % sicher und trotzdem komplett falsch sein; das siehst du sofort, wenn du ihm etwas zeigst, das gar nicht im Katalog steht.

Frage 2: Dein Bürostuhl wird zuverlässig erkannt, dein 3D-Drucker daneben nie. Warum?

- Der 3D-Drucker ist zu klein und zu dunkel für jedes Kameramodell.
- Das Modell muss für neue Geräte erst online nachfragen, was ohne Server nicht geht.
- **✓ „chair“ steht im 80-Klassen-Katalog von COCO (2017), „3D-Drucker“ nicht — was nicht im Katalog steht, kann nicht ausgegeben werden.**

Die Ausgabeschicht hat exakt 80 Ausgänge, einen pro Klasse. Für „3D-Drucker“ gibt es dort schlicht keinen Platz. Willst du eigene Kategorien, musst du das Netz nachtrainieren — genau darum geht es im späteren Kapitel „eigene Bild-KI“ (Transfer Learning).

Frage 3: Was liefert das Netz als Rohausgabe zurück?

- Einen deutschen Satz, der die Szene beschreibt.
- **✓ Zahlen: Rahmen-Koordinaten (x, y, Breite, Höhe), Punktzahlen und Klassennummern — die Rahmen und Wörter malen wir selbst dazu.**
- Ein fertiges Bild mit eingezeichneten farbigen Rahmen.

Rein geht ein Tensor [1, Höhe, Breite, 3] mit Werten 0–255, raus kommen zwei Zahlenlisten: Boxen und Scores. Farben, Beschriftung, deutsche Wörter und der Prozentwert sind reine Darstellung — sechs Zeilen Canvas-Code in modul.js.

Das nimmst du mit

- **Klassifikation fragt „was“, Detektion fragt „was *wo*“.** Ein Wort fürs ganze Bild gegen null bis zwanzig Funde mit Rahmen, Etikett und Punktzahl.
- **Ein Detektor ist ein Klassifikator mit zweitem Kopf.** Derselbe Faltungs-Turm, ein Ausgang rät die Klasse, der andere die vier Zahlen des Rahmens — „Single Shot“ heißt: ein Durchlauf, alle Funde auf einmal.
- **Die Rohausgabe sind nur Zahlenlisten.** Rahmen-Koordinaten, Punktzahlen, Klassennummern — die bunten Kästchen malt erst dein Browser dazu.
- **Die Prozentzahl ist keine Wahrheit.** „87 %“ heißt „von meinen 80 Schubladen passt diese am besten“ — dieselbe Logik, die bei Sprachmodellen zu Halluzinationen führt.
- **80 Klassen, eingefroren.** Bürostuhl ja, 3D-Drucker nein — und ungewohnte Blickwinkel, schlechtes Licht und kleine Objekte lassen die Punktzahlen einbrechen.

Weiter geht’s: Das Netz von heute ist der große Bruder deines selbst trainierten Ziffern-CNN aus „Ziffern erkennen“ — gleiche Filter-Idee, nur mit einem zweiten Ausgang für die Rahmen. Warum „87 %“ trotzdem keine Wahrheit ist, steht in „Halluzinationen“. Und als Nächstes sieht die KI nicht nur, *dass* du da bist, sondern *wie* du stehst: Im Pose-Kapitel (440) werden aus dem einen Rahmen 17 Körperpunkte.

So zitierst du dieses Kapitel

HerrKI: Was sieht die KI? (Objekterkennung). herrki.de/430_objekterkennung, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.