Dein Körper als Tensor (Pose)

Kapitel online

Deine Kamera oder ein Foto — und du siehst live, was eine Pose-KI wirklich von dir sieht: 17 Punkte, 51 Zahlen, ein [17, 3]-Tensor. Mit Skelett-Overlay und einem Hampelmann-Zähler, der nur aus diesen Zahlen gebaut ist.

Interaktiv · Kapitel 46 von 50 · ca. 20 Min. · 300 XP

Einfach gesagt: Deine Kamera sieht dich als 17 Punkte an deinem Körper, nicht als Person — daraus zählt ein kurzes Stück Code deine Hampelmänner.

Deine Fitness-App zählt Kniebeugen und meckert, wenn der Rücken krumm ist. Eine Spielkonsole lässt dich ohne Controller Tennis spielen — dein Körper *ist* der Controller. Eine Physiotherapie-App prüft zu Hause, ob du die Übung wirklich richtig machst. Und in jedem Handy-Foto weiß die Galerie schon, wer da wie dasteht.

Das wirkt wie Zauberei: Die Software „sieht“ deinen Körper. Heute schauen wir nach, was da wirklich passiert — und die Antwort ist herrlich unspektakulär. Die KI sieht keinen Körper, keine Bewegung, keine Anstrengung. Sie sieht **51 Zahlen**. Mehr nicht. Und du siehst sie gleich **live neben deinem eigenen Bild**.

Was du hier lernst

- Wie aus einem Kamerabild **[Höhe, Breite, 3]** ein winziger **[17, 3]-Tensor** wird
- Was die 17 Zeilen und die 3 Spalten *genau* bedeuten — Gelenk, x, y, Sicherheit
- Warum y nach **unten** größer wird (und warum dich das sonst dauernd verwirren würde)
- Wie aus diesen Zahlen in **drei Zeilen Logik** ein Hampelmann-Zähler wird — ganz ohne neues Training
- Wo das Verfahren ehrlich an seine Grenzen stößt

Für den Unterricht

Einstieg (4 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (12 Min): die interaktiven Module auf herrki.de/440_koerper_als_tensor ausprobieren.

Sichern mit Quiz (4 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Deine Daten bleiben auf deinem Gerät

Gleich siehst du dich selbst auf dem Bildschirm — deshalb vorweg, in aller Klarheit: **Dein Kamerabild verlässt dein Gerät nicht.** Kein Bild, kein Video, kein einzelnes Skelett wird irgendwohin hochgeladen. Es gibt in diesem Kapitel keine einzige Zeile Code, die etwas an einen Server schickt.

- Der Browser liefert das Kamerabild über `getUserMedia` direkt an die Seite — wie bei einem Video-Chat, nur ohne Gegenstelle.
- Das Pose-Modell (MoveNet, 4,8 MB) liegt **auf diesem Server** und rechnet danach in **deinem** Browser, auf deinem Prozessor bzw. deiner Grafikkarte.
- Nichts wird gespeichert. Tab schließen = alles weg. Der Kamera-Stream wird beim Stoppen sauber abgeschaltet (die Kamera-Leuchte geht aus).

Du willst die Kamera gar nicht anschalten? Völlig in Ordnung — **der Foto-Weg unten funktioniert genauso gut**, und auch dein Foto bleibt lokal.

Schritt 1: Das Werkzeug holen

Am Gerät ausprobieren: herrki.de/440_koerper_als_tensor#abschnitt-3

Direkt zum Modul



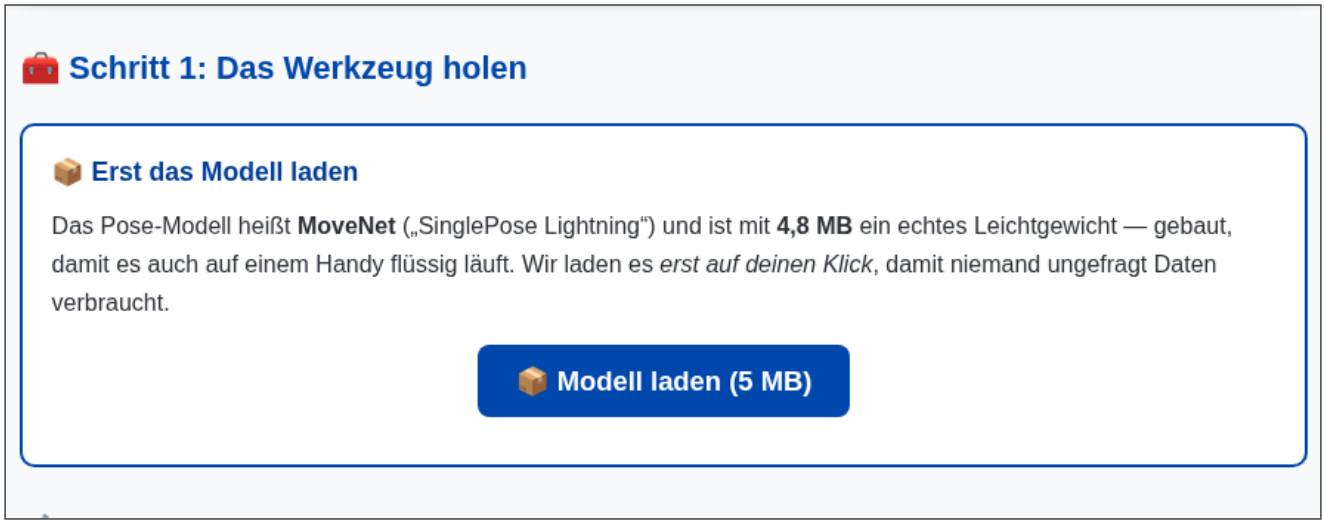
Interaktives Modul – am Gerät ausprobieren

Schritt 2: Das Skelett — und was wirklich dahintersteckt

Links siehst du dich (oder dein Foto) mit dem Skelett darüber: **17 Punkte** und die Linien dazwischen. Rechts steht dasselbe als das, was es in Wahrheit ist — eine **Tabelle aus Zahlen**. Beides ist dieselbe Information. Nur die linke Seite ist für dich gemalt.

Am Gerät ausprobieren: herrki.de/440_koerper_als_tensor#abschnitt-4

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Bereit.

[17, 3] **Das ist ALLES, was die KI von dir sieht: ein [17, 3]-Tensor**

#	Gelenk	x	y	Sicherheit

x und **y** sind Pixel-Positionen im Originalbild, **Sicherheit** ist ein Wert zwischen 0 und 1. Zeilen unter **0,30** sind ausgegraut — die zeichnen wir gestrichelt oder gar nicht.

Runter zum Tensor: aus 6 Millionen Zahlen werden 51

Rechnen wir das einmal durch. Ein Kamerabild in Full-HD ist ein Tensor der Form **[1080, 1920, 3]** — Höhe mal Breite mal drei Farbkanäle (Rot/Grün/Blau, genau wie in „Bilder als Tensoren“). Das sind **6.220.800 Zahlen**. Sechs Millionen. Pro Einzelbild.

[1080, 1920, 3] **Kamerabild** 6,2 Mio. Helligkeitswerte — für einen Computer erstmal komplett bedeutungslos

↓ MoveNet (ein neuronales Netz, 4,8 MB Gewichte)

[192, 192, 3] **zurechtgeschnitten & verkleinert** Das Netz schaut sich immer nur einen kleinen Ausschnitt in fester Größe an

↓ Faltungen, Schicht für Schicht

[17, 3] **17 Gelenke × (x, y, Sicherheit)** 51 Zahlen — die komplette „Wahrnehmung“ deines Körpers

Der Kernsatz des Kapitels: „Sehen“ heißt hier nichts anderes als **6 Millionen Pixel-Zahlen werden zu 51 Zahlen eingedampft** — und mit 51 Zahlen kann ganz normale Programmlogik arbeiten. Kein Zauber, keine Magie: eine Kompression auf das, worauf es ankommt.

Genau *das* ist der Grund, warum eine Fitness-App auf deinem Handy flüssig läuft. Nicht das Netz macht die Kniebeugen-Zählung — das Netz liefert nur die 51 Zahlen. Gezählt wird danach, mit stinknormalem Code. Und den bauen wir jetzt.

Schritt 3: Vom Tensor zur Anwendung — der Hampelmann-Zähler

Ein Hampelmann (englisch: *jumping jack*) ist eine sehr einfache Bewegung: Arme hoch, Arme runter, hoch, runter. Wie zählt man die? Man braucht **kein** neues neuronales Netz, **kein** Training, **keine** Trainingsdaten. Man braucht drei Zeilen Logik auf dem Tensor, den wir schon haben:

```
// Beide Handgelenke ÜBER der Nase? -> Arme sind oben.
const hoch = handLinks.y < nase.y && handRechts.y < nase.y;
// Beide Handgelenke UNTER den Hüften? -> Arme sind unten.
const runter = handLinks.y > hueftLinks.y && handRechts.y > hueftRechts.y;
// Nur der Wechsel von UNTEN nach OBEN zählt einen Hampelmann.
if (hoch && zustand === 'runter') { zaehler = zaehler + 1; }
```

Moment — „über der Nase“ heißt kleiner?

Ja, und das ist die klassische Stolperfalle. In der Mathematik-Schule wächst `y` nach **oben**. In Bildern wächst `y` nach **unten**: Zeile 0 ist die *oberste* Bildzeile, danach zählt man abwärts. Der Grund ist uralt und banal — Bilder werden seit jeher zeilenweise von oben nach unten gespeichert und übertragen, genau wie du diesen Text liest.

Für uns heißt das: **weiter oben im Bild = kleineres y**. Deshalb bedeutet `handgelenk.y < nase.y` genau: „Die Hand ist *höher* als die Nase.“ Schau in der Tabelle oben nach — die Nase hat ein kleineres y als die Knöchel. Immer.

Warum überhaupt der Umweg über zwei Zustände (`hoch` / `runter`) und nicht einfach „Hände oben = +1“? Weil du sonst pro Sekunde 15-mal weiterzählen würdest, solange die Arme oben bleiben. Wir zählen deshalb nur den **Wechsel** — das ist der gleiche Trick, mit dem eine Türklingel zwischen „gedrückt“ und „gerade eben gedrückt worden“ unterscheidet.

0

Hampelmänner

Am Gerät ausprobieren: herrki.de/440_koerper_als_tensor#abschnitt-6

Direkt zum Modul

🏃 Schritt 3: Vom Tensor zur Anwendung — der Hampelmann-Zähler

Ein Hampelmann (englisch: *jumping jack*) ist eine sehr einfache Bewegung: Arme hoch, Arme runter, hoch, runter. Wie zählt man die? Man braucht **kein** neues neuronales Netz, **kein** Training, **keine** Trainingsdaten. Man braucht drei Zeilen Logik auf dem Tensor, den wir schon haben:

```
// Beide Handgelenke ÜBER der Nase? -> Arme sind oben.
const hoch   = handLinks.y < nase.y && handRechts.y < nase.y;
// Beide Handgelenke UNTER den Hüften? -> Arme sind unten.
const runter = handLinks.y > hueftLinks.y && handRechts.y > hueftRechts.y;
// Nur der Wechsel von UNTEN nach OBEN zählt einen Hampelmann.
if (hoch && zustand === 'runter') { zaehler = zaehler + 1; }
```

⚠️ Moment — „über der Nase“ heißt *kleiner*?

Ja, und das ist die klassische Stolperfalle. In der Mathematik-Schule wächst `y` nach **oben**. In Bildern wächst `y` nach **unten**: Zeile 0 ist die *oberste* Bildzeile, danach zählt man abwärts. Der Grund ist uralt und banal — Bilder werden seit jeher zeilenweise von oben nach unten gespeichert und übertragen, genau wie du diesen Text liest.

Für uns heißt das: **weiter oben im Bild = kleineres y**. Deshalb bedeutet `handgelenk.y < nase.y` genau: „Die Hand ist *höher* als die Nase.“ Schau in der Tabelle oben nach — die Nase hat ein kleineres y als die Knöchel. Immer.

Warum überhaupt der Umweg über zwei Zustände (`hoch` / `runter`) und nicht einfach „Hände oben = +1“? Weil du sonst pro Sekunde 15-mal weiterzählen würdest, solange die Arme oben bleiben. Wir zählen deshalb nur den **Wechsel** — das ist der gleiche Trick, mit dem eine Türklingel zwischen „gedrückt“ und „gerade eben gedrückt worden“ unterscheidet.

0

HAMPELMÄNNER

Zustand: – (noch keine Person erkannt)

↺ Zähler zurücksetzen

Funktioniert mit der Live-Kamera. Für ein Foto zeigt der Zähler nur den erkannten Zustand an — ein Standbild kann sich schlecht bewegen. 🙂

🚀 **Und genau so funktionieren die echten Apps.** Kniebeugen-Zähler: Winkel im Knie beobachten. Haltungs-Warner: Abstand Schulter–Ohr messen. Spielkonsolen-Steuerung: prüfen, in welchem Bildbereich die Handgelenke liegen. Immer dasselbe Muster — **das Netz liefert den Tensor, die Anwendung ist Geometrie darauf.**

Interaktives Modul – am Gerät ausprobieren

Zustand: — (noch keine Person erkannt)

Funktioniert mit der Live-Kamera. Für ein Foto zeigt der Zähler nur den erkannten Zustand an — ein Standbild kann sich schlecht bewegen.

Und genau so funktionieren die echten Apps. Kniebeugen-Zähler: Winkel im Knie beobachten. Haltungs-Warner: Abstand Schulter–Ohr messen. Spielkonsolen-Steuerung: prüfen, in welchem Bildbereich die Handgelenke liegen. Immer dasselbe Muster — **das Netz liefert den Tensor, die Anwendung ist Geometrie darauf.**

Ehrlich bleiben: was das hier NICHT kann

- **Alles ist flach.** Die 51 Zahlen enthalten *keine Tiefe*. Es gibt nur x und y — wo im Bild, nicht wie weit weg. Dreh dich seitlich zur Kamera, und das Modell rät: Die Schulter, die eigentlich hinten ist, bekommt trotzdem eine Bildposition. Für „ist der Rücken gerade?“ ist das oft zu wenig.
- **Verdeckung verwirrt.** Was das Modell nicht sieht, muss es erraten — Hand hinter dem Rücken, Beine hinter einem Tisch. Die Sicherheits-Spalte sinkt dann, aber sie sinkt nicht immer ehrlich genug.
- **Licht und Kontrast entscheiden.** Dunkler Raum, Gegenlicht, dunkle Kleidung vor dunklem Hintergrund: Die Sicherheiten brechen ein. Probier's aus — geh mal zwei Schritte in den Schatten und beobachte die rechte Spalte.
- **Genau eine Person.** Wir nutzen „SinglePose“ — das Modell sucht *eine* Person und liefert immer 17 Punkte, auch wenn zwei Leute im Bild sind (dann bastelt es gern ein Misch-Skelett). Für mehrere Personen gibt es größere Varianten, die auch mehr Rechenleistung kosten.
- **Kein Verständnis von Bewegung.** Das ist der wichtigste Punkt: Die KI weiß nicht, dass du Sport machst. Sie weiß nicht einmal, dass sie einen Menschen sieht. Sie ordnet einem Bildausschnitt 17 Koordinaten zu — **der Rest ist Geometrie auf 51 Zahlen**, geschrieben von einem Menschen, der weiß, was ein Hampelmann ist.

Zurück zur Anwendung

Zurück zur Fitness-App, die Kniebeugen zählt, zur Spielkonsole ohne Controller und zur Physio-App, die deine Haltung prüft: Keine davon „sieht“ dich. Alle bekommen dieselben 51 Zahlen wie du gerade — und was daraus eine Kniebeuge oder ein Hampelmann wird, hat ein Mensch als Geometrie darübergelegt. Genau das kannst du jetzt auch: Gelenkpunkte auslesen, die y-Achse richtig herum denken, die Sicherheits-Spalte prüfen und mit drei Zeilen Vergleichslogik eine eigene Übung zählen.

Und weil das Muster dahinter immer dasselbe ist, wechseln wir im nächsten Kapitel „Sprich mit deinem Browser (Audio)“ einfach den Sinn: Statt Bildpunkten kommt Klang herein — der erst zu einem Bild wird, bevor ihn dieselbe Art Netz erkennt.

Zum Weiterlesen

- Cao u. a. 2018, *„OpenPose: Realtime Multi-Person 2D Pose Estimation“* — arxiv.org/abs/1812.08008 — der Klassiker für *mehrere* Personen gleichzeitig, also genau die Grenze, an die wir hier gestoßen sind.
- Bazarevsky u. a. 2020, *„BlazePose: On-device Real-time Body Pose Tracking“* — arxiv.org/abs/2006.10204 — wie man ein Pose-Netz so klein bekommt, dass es auf dem Handy in Echtzeit läuft.
- TensorFlow Blog 2021, *Next-Generation Pose Detection with MoveNet and TensorFlow.js* — blog.tensorflow.org (blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html) — die Vorstellung genau des Modells, das dir hier gerade dein Skelett gezeichnet hat.

Kurzer Check – mit Lösung

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

Frage 1: Was liefert MoveNet als Rohausgabe, wenn du ein Bild hineingibst?

- Ein freigestelltes Bild des Körpers ohne Hintergrund.
- **✓ Einen [17, 3]-Tensor: 17 Gelenke mit je x, y und einem Sicherheitswert — 51 Zahlen.**
- Einen Text wie „Person steht mit erhobenen Armen“.

17 Zeilen (Nase, Augen, Ohren, Schultern, Ellbogen, Handgelenke, Hüften, Knie, Knöchel) mal 3 Spalten (x, y, Sicherheit). Das gezeichnete Skelett ist nur unsere Darstellung dieser Zahlen — das Modell selbst malt nichts.

Frage 2: Der Hampelmann-Zähler funktioniert ohne jedes zusätzliche Training. Warum?

- Weil MoveNet insgeheim schon auf Hampelmänner trainiert wurde.
- Weil der Browser die Bewegung selbst erkennt und meldet.
- **✓ Weil die Regel als normale Logik AUF dem Tensor läuft: Handgelenke mit Nase und Hüften vergleichen — dafür braucht es kein neues Netz.**

Das Netz erledigt den schweren Teil (6 Mio. Pixel → 51 Zahlen). Der leichte Teil danach ist gewöhnliche Programmierung. Ein Kniebeugen-Zähler wäre dieselbe Bauart mit einer anderen Regel — wieder ohne Training.

Frage 3: In der Tabelle hat die Nase ein kleineres y als die Knöchel. Warum ist „oben“ das kleinere y?

- Weil die Nase kleiner ist als ein Fuß.
- **✓ Weil Bildkoordinaten oben links beginnen: Zeile 0 ist die oberste Bildzeile, y wächst nach unten.**
- Weil MoveNet die Werte nach der Sicherheit sortiert.

Bilder werden zeilenweise von oben nach unten gespeichert — der Ursprung (0,0) liegt oben links, nicht wie im Mathe-Heft unten links. Deshalb heißt „Hand höher als Nase“ im Code `hand.y < nase.y`.

Das nimmst du mit

- **Aus Millionen Zahlen werden 51.** Ein Full-HD-Bild ist ein **[1080, 1920, 3]**-Tensor mit 6.220.800 Werten — die Antwort des Netzes ist ein **[17, 3]**-Tensor.
- **17 Zeilen, 3 Spalten — mehr steht da nicht.** Pro Gelenk eine Zeile, darin x, y und die Sicherheit des Modells.
- **y wächst nach unten.** Bilder werden von oben links gespeichert; „Hand höher als Nase“ heißt im Code deshalb `hand.y < nase.y`.
- **Der Hampelmann-Zähler ist Geometrie, kein Training.** Ein paar Zeilen Logik auf 51 Zahlen — geschrieben von einem Menschen, der weiß, was ein Hampelmann ist.
- **Alles ist flach, und Verdeckung verwirrt.** Keine Tiefe, genau eine Person, und bei Gegenlicht oder dunkler Kleidung brechen die Sicherheiten ein.

Weiter geht’s: In „Was sieht die KI? (Objekterkennung)“ hat ein Netz Kästen um Dinge gemalt — hier hat dieselbe Maschinerie *Punkte an deinem Körper* ausgegeben: gleicher Bauplan, anderer Ausgabe-Tensor. **Als Nächstes hört dein Browser aufs Wort** — im Audio-Kapitel (450) siehst du, dass sogar Schall bei uns als Tensor ankommt. Und danach trainierst du in Kapitel 460 endlich selbst eine Bild-KI — auf deinen eigenen Bildern.

So zitierst du dieses Kapitel

HerrKI: Dein Körper als Tensor (Pose). herrki.de/440_koerper_als_tensor, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.