Bilder als Tensoren

Wie RGB-Bilder zu Zahlen werden und warum alle Bilder gleich groß sein müssen.

Grundlagen · Kapitel 6 von 50 · ca. 15 Min. · 100 XP

Kapitel online

Einfach gesagt: Dein Selfie ist für den Computer nur ein Würfel aus Zahlen: für jedes Pixel ein Rot-, Grün- und Blau-Wert von 0 bis 255.

Stell dir vor: Du machst ein Foto mit deinem Handy - ein Selfie, deine Katze, eine Landschaft. Für dich ist das ein Bild. Aber für den Computer? **Nur ein Haufen Zahlen!**

Heute zeige ich dir, wie Computer Bilder "sehen": Jedes Foto wird in einen **3D-Tensor** (Rank 3 aus Kapitel „Was sind Tensoren?") umgewandelt - eine riesige Tabelle voller RGB-Werte (Rot-Grün-Blau). Das ist die Grundlage für alles: Gesichtserkennung, Instagram-Filter, selbstfahrende Autos - alles basiert darauf!

Das Coole daran: Sobald du verstehst, wie Bilder zu Tensoren werden, kannst du eigene Bilderkennungs-Modelle bauen - ob Katzen vs. Hunde erkennen, Hausnummern lesen oder Gesichter identifizieren. Alles startet mit RGB-Zahlen!

HerrKI-Tipp: Ein Bild ist ein 3D-Würfel aus Zahlen! **Höhe × Breite × 3 RGB-Kanäle**. Ein 200×200 Pixel Foto = 120.000 Zahlen. Krass, oder?

Was du hier lernst

- Wie Bilder in **RGB-Zahlen (0-255)** umgewandelt werden
- Warum Bilder **3D-Tensoren** sind (Höhe × Breite × 3 Farben)
- Wie du mit **TensorFlow.js** Bilder einliest und verarbeitest
- Warum neuronale Netze **gleich große Bilder** brauchen

Für den Unterricht

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

Erkunden am Gerät (9 Min): die interaktiven Module auf herrki.de/020_bilder_als_tensoren ausprobieren.

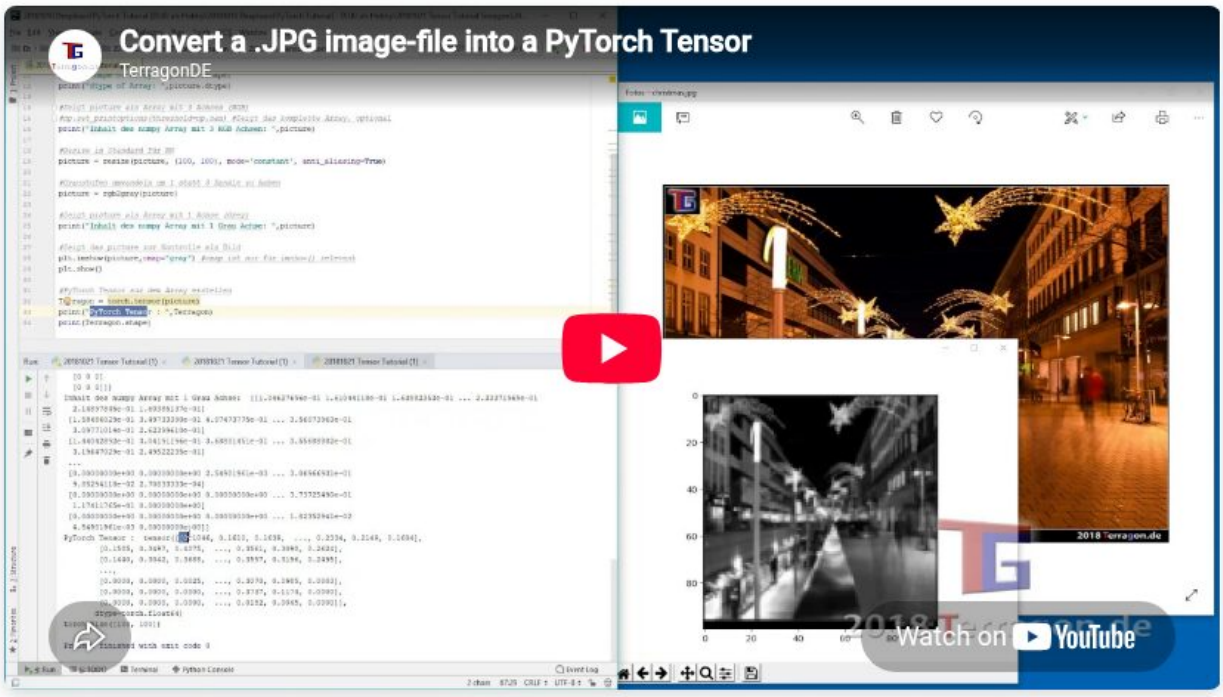
Sichern mit Quiz (3 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Video-Erklärung von HerrKI

Video ansehen: herrki.de/020_bilder_als_tensoren#abschnitt-2

Direkt zum Modul

Schau dir dieses Video an, um Bilder als Tensoren noch besser zu verstehen:

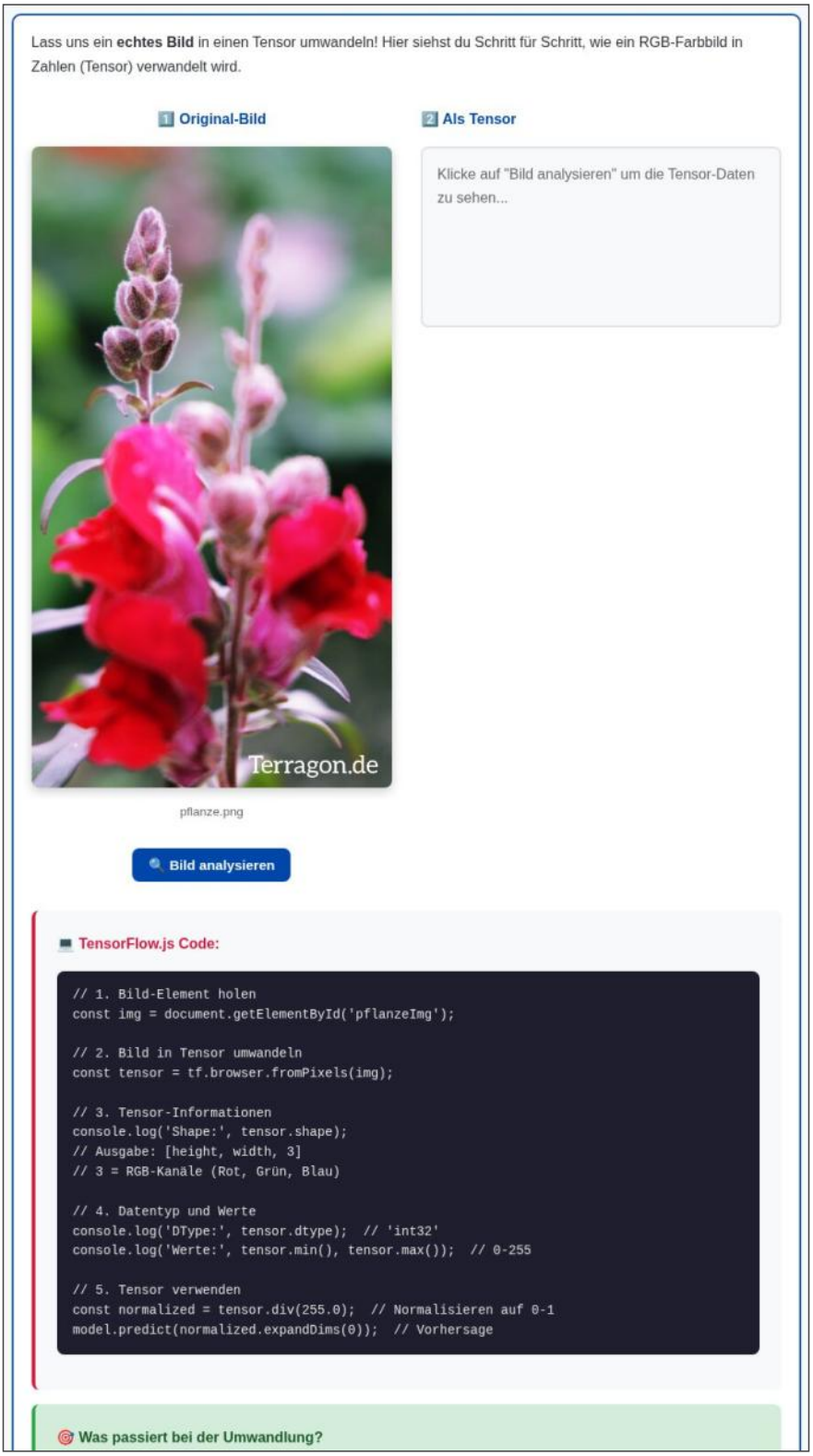


Video auf der Kapitelseite – am Gerät ansehen

Runter zum Tensor: RGB-Bild Tensor

Am Gerät ausprobieren: herrki.de/020_bilder_als_tensoren#abschnitt-3

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Einheitliche Bildgröße für Neuronale Netze

Am Gerät ausprobieren: herrki.de/020_bilder_als_tensoren#abschnitt-4

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Zurück zur Anwendung

Und damit schließt sich der Kreis: Wenn Instagram deinem Foto einen Filter verpasst oder dein Handy „Katze“ in deine Galerie schreibt, läuft genau diese Kette — Foto → Zahlen-Würfel (H × B × 3) → Mathe auf dem Tensor → Ergebnis. Der Filter ist dabei nichts anderes als Rechnen auf den 120.000 Zahlen deines Selfies, und „Katze“ ist die größte Zahl, die am Ende aus dem Netz fällt. Und DESHALB will jede Bild-KI deine Fotos zuerst auf eine feste Größe bringen: Erst wenn alle Würfel dieselbe Shape haben, passen sie in dasselbe Netz.

Als Nächstes nehmen wir uns den Datentyp vor, aus dem die meisten echten Projekte bestehen: Excel- und CSV-Tabellen — deutlich weniger Zahlen, dafür mit Bedeutung in jeder Spalte.

Zum Weiterlesen

- **Die Vorlesung dazu:** Stanford CS231n, *Image Classification* — cs231n.github.io/classification — erklärt, warum „Katze erkennen“ für einen Zahlen-Würfel so viel schwerer ist, als es aussieht (auf Englisch)
- **Woher die Pixel kommen:** MDN, *ImageData* — developer.mozilla.org/de/docs/Web/API/ImageData — das RGBA-Array des Browsers, aus dem TensorFlow.js den Tensor baut (auf Deutsch)
- **Die Funktion im Original:** TensorFlow.js-API, *tf.browser.fromPixels* — js.tensorflow.org/api/latest/#browser.fromPixels — alle Parameter der einen Zeile, die aus deinem Bild einen Tensor macht

Kurzer Check – mit Lösung

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

Drei schnelle Fragen, dann bist du durch — klick einfach deine Antwort an:

Frage 1: Warum ist ein Farbbild ein 3D-Tensor?

- Weil Bilder räumlich wirken
- **✔ Höhe × Breite × 3 Farbkanäle (R, G, B) — jedes Pixel bringt drei Zahlen mit**
- Weil man drei Bilder gleichzeitig laden muss

Richtig: Ein 100×100-Farbbild hat die Shape [100, 100, 3] — für jedes der 10.000 Pixel je ein Rot-, Grün- und Blau-Wert. Zusammen 30.000 Zahlen!

Frage 2: Welche Werte kann ein einzelner Farbkanal eines Pixels annehmen?

- **✔ 0 bis 255 — und fürs Training teilt man oft durch 255**
- Nur 0 oder 1 (schwarz oder weiß)
- Beliebig negative und positive Werte

Genau: 0 = ganz dunkel, 255 = volle Intensität. Mit tensor.div(255.0) quetscht man die Werte auf 0–1 — damit trainiert das Netz stabiler.

Frage 3: Warum müssen alle Bilder für ein neuronales Netz gleich groß sein?

- Damit die Galerie ordentlich aussieht
- **✔ Weil die Eingabe-Shape des Netzes festliegt — die Anzahl der Gewichte steht nach dem Bau fest**
- Weil TensorFlow sonst zu langsam wird

Das Netz ist wie ein Formular mit fester Felderzahl: Für jede Eingabezahl gibt es Gewichte. Deshalb werden Bilder vor dem Training auf eine einheitliche Größe skaliert (z. B. 224×224).

Das nimmst du mit

- **Ein Bild ist ein Zahlen-Würfel.** Shape `(Höhe, Breite, 3)` — ein 200×200-Foto sind 120.000 Zahlen.
- **Jedes Pixel bringt drei Werte mit:** Rot, Grün, Blau — jeder von 0 (dunkel) bis 255 (volle Intensität).
- **Eine Zeile macht den Tensor:** `tf.browser.fromPixels(img)` — Bild-Element rein, Tensor raus.
- **Normalisieren heißt „durch 255“.** `tensor.div(255.0)` bringt alle Werte auf 0.0–1.0, damit das Training stabil läuft.
- **Alle Bilder müssen gleich groß sein.** Die `inputShape` des Netzes steht fest — also vorher skalieren oder zuschneiden (224×224, 28×28, 32×32).

So zitierst du dieses Kapitel

HerrKI: Bilder als Tensoren. herrki.de/020_bilder_als_tensoren, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.