Sprich mit deinem Browser (Audio)

Kapitel online

Dein Browser hört zu — und versteht: Aus einer Sekunde Ton wird ein Spektrogramm, aus dem Spektrogramm ein Bild, aus dem Bild ein Wort. Steuere eine Spielfigur per Zuruf und trainiere dem Modell danach dein eigenes Wort an. Kein Ton verlässt dabei dein Gerät.

Interaktiv · Kapitel 47 von 50 · ca. 25 Min. · 300 XP

Einfach gesagt: Dein Browser malt aus einer Sekunde Sprache ein Bild aus Tonhöhen und erkennt darin dein gesprochenes Wort — dein Ton bleibt auf deinem Gerät.

„Alexa, mach das Licht an.“ „Hey Siri, stell einen Wecker.“ Millionen Menschen sprechen jeden Tag mit Maschinen, und die meisten haben sich nie gefragt, was in der Sekunde zwischen Zuruf und Reaktion eigentlich passiert. Wie *hört* ein Computer? Er hat keine Ohren, kein Gehirn und keine Ahnung von Sprache — er hat nur Zahlen.

Heute holst du dir die Antwort in deinen eigenen Browser. Kein Cloud-Dienst, kein Konto, kein Kabel nach Seattle: Ein 6 Megabyte kleines Modell landet auf deinem Gerät, hört dir zu und versteht 18 gesprochene Kommandos. Damit steuerst du gleich eine Spielfigur. Und danach machst du das, was sonst nur Forschungsteams machen: Du bringst dem Modell **ein eigenes, neues Wort** bei.

Roter Faden: Wir gehen von der Sprachassistentin, die jeder kennt, runter bis zum Tensor — und wieder hoch bis zu einem Modell, das *dein* Wort kennt. Am Ende weißt du genau, warum dein Handy „stop“ versteht und „Stopp“ nicht.

Was du hier lernst

- Warum Spracherkennung in Wahrheit **Bild-Erkennung** ist — der wichtigste Aha-Moment des Kapitels
- Was ein **Spektrogramm** ist und wie aus 1 Sekunde Klang der Tensor `[43, 232, 1]` wird
- Dass am Ende auch hier nur eine nüchterne Zahlenreihe steht — 20 Werte, mehr sagt das Netz nicht
- Wie **Transfer-Learning** funktioniert: großes Modell einfrieren, winzigen Kopf antrainieren
- Und ganz praktisch: eine Spielfigur per Zuruf steuern und dem Modell dein eigenes Wort beibringen

Für den Unterricht

Einstieg (5 Min): Kapitel gemeinsam lesen, Alltagsbeispiel aufgreifen.

Erkunden am Gerät (15 Min): die interaktiven Module auf herrki.de/450_audio_kommandos ausprobieren.

Sichern mit Quiz (5 Min): „Kurzer Check" besprechen, „Das nimmst du mit" festhalten.

Dein Mikrofon-Ton bleibt auf deinem Gerät

Das ist der eigentliche Clou dieses Kapitels — und der Unterschied zu jeder Sprachassistentin, die du kennst. Bei Alexa, Siri und Google Assistant wird der aufgenommene Ton (oder eine daraus berechnete Darstellung) **zu einem Rechenzentrum geschickt**, dort erkannt und beantwortet. Deshalb gab es die Schlagzeilen über Mitarbeiter, die Sprachaufnahmen zur Qualitätskontrolle anhören.

Hier läuft es anders herum:

- Das **Modell kommt zu dir** — einmalig 6 MB, dann liegt es in deinem Browser.
- Dein Ton geht nur an die Web-Audio-Schnittstelle deines Browsers und von dort direkt in TensorFlow.js. Es gibt in diesem Kapitel **keine einzige Zeile Code, die etwas an einen Server schickt**.
- Nichts wird gespeichert. Seite neu laden = alles weg, auch dein selbst antrainiertes Wort.

Wenn du misstrauisch bist (sehr gesund!): Entwicklertools öffnen, Reiter „Netzwerk“, dann sprechen. Du siehst dort das einmalige Laden der Modelldatei — und danach nichts mehr. Kein Byte verlässt dein Gerät, solange du redest.

Bevor du loslegst: drei ehrliche Hinweise

- **Du brauchst ein Mikrofon.** Ein Handy ist dafür sogar ideal — halte es beim Sprechen etwa eine Handbreit vor den Mund. Der Browser fragt dich einmal um Erlaubnis; ohne dein „Ja“ passiert nichts. (Ohne Mikrofon kommst du trotzdem durch: Es gibt weiter unten einen Datei-Notnagel und Ersatzknöpfe fürs Spiel.)
- **Das Modell versteht 18 *englische* Wörter** — es wurde auf englischen Sprechern trainiert. „Stopp“ auf Deutsch wird es also eher nicht erkennen, „stop“ auf Englisch schon. Das ist keine Macke, sondern die ehrliche Konsequenz aus den Trainingsdaten (siehe „Daten vorbereiten“).
- **Auf iPhone und iPad** darf ein Browser erst nach einem echten Fingertipp aufs Mikrofon zugreifen — deshalb startet hier alles per Knopfdruck und nicht automatisch beim Seitenaufruf.

Das komplette Vokabular (blau = die Steuerwörter fürs Spiel)

yes — ja

no — nein

zero — null

one — eins

two — zwei

three — drei

four — vier

five — fünf

six — sechs

seven — sieben

eight — acht

nine — neun

Das Netz kennt noch zwei weitere Kategorien, die keine Wörter sind: `_background_noise_` (Hintergrundgeräusch) und `_unknown_` (irgendein Wort, das nicht dazugehört). Zusammen sind das **20 Ausgänge** — merk dir die Zahl, sie kommt gleich wieder.

Runter zum Tensor: Der Kern-Trick — Hören ist hier Bild-Erkennung

Jetzt kommt der Gedanke, wegen dem sich dieses Kapitel lohnt. Man würde vermuten, dass Spracherkennung etwas völlig Eigenes ist — Phoneme, Grammatik, Wörterbücher. Ist sie hier **nicht**. Der Browser macht mit deinem Ton etwas verblüffend Simples: **Er malt ihn.**

Ein Mikrofon misst nur eine einzige Größe: den Luftdruck, sehr oft hintereinander — 44.100-mal pro Sekunde. Eine Sekunde Ton ist also erstmal ein langer Vektor mit 44.100 Zahlen. Damit kann ein Bilderkennungs-Netz nichts anfangen. Also wird der Ton in **43 kleine Zeitfenster** von je 23 Millisekunden zerschnitten, und für jedes Fenster fragt eine **Fourier-Transformation** (FFT): *Welche Tonhöhen stecken in diesem Moment drin, und wie laut?* Die Antwort sind 232 Zahlen — eine pro Frequenzband, von tief bis hoch.

43 Zeitschritte × 232 Frequenzen: Das ist eine Tabelle aus Zahlen — und eine Tabelle aus Zahlen ist genau das, was du aus „Bilder als Tensoren“ kennst. **Ein Spektrogramm ist ein Bild.** Waagerecht die Zeit, senkrecht die Tonhöhe, die Helligkeit ist die Lautstärke. Und deshalb steckt hinter der Spracherkennung hier dasselbe **CNN** — dasselbe Faltungs-Prinzip mit Filtern, die Kanten und Muster suchen — wie bei den handgeschriebenen Ziffern in „Ziffern erkennen“. Nur ist das Bild diesmal ein Klang.

```
// Die Reise von 1 Sekunde Zuruf bis zur Antwort — die Shapes
Mikrofon [44100] // Luftdruck, 44.100-mal pro Sekunde gemessen
     ↓ in 43 Fenster à 23 ms schneiden, je eine FFT
Spektrogramm [43, 232, 1] // 43 Zeitschritte × 232 Frequenzen × 1 Kanal = ein BILD
     ↓ 4 Faltungsschichten + Pooling (dasselbe Prinzip wie in Kapitel 420)
Merkmale [2000] // was das Netz aus dem Bild herausgelesen hat
     ↓ eine letzte Dense-Schicht + Softmax
Scores [20] // 20 Wahrscheinlichkeiten, zusammen genau 1,0
```

Hier siehst du dieses Bild gleich live. Sobald du unten das Mikrofon startest, malt der Browser das Spektrogramm des zuletzt erkannten Wortes in dieses Feld — links der Anfang der Sekunde, rechts das Ende, unten die tiefen Töne, oben die hohen. Hell = laut.

Am Gerät ausprobieren: herrki.de/450_audio_kommandos#abschnitt-4

Direkt zum Modul

Jetzt kommt der Gedanke, wegen dem sich dieses Kapitel lohnt. Man würde vermuten, dass Spracherkennung etwas völlig Eigenes ist — Phoneme, Grammatik, Wörterbücher. Ist sie hier **nicht**. Der Browser macht mit deinem Ton etwas verblüffend Simples: **Er malt ihn.**

Ein Mikrofon misst nur eine einzige Größe: den Luftdruck, sehr oft hintereinander — 44.100-mal pro Sekunde. Eine Sekunde Ton ist also erstmal ein langer Vektor mit 44.100 Zahlen. Damit kann ein Bilderkennungs-Netz nichts anfangen. Also wird der Ton in **43 kleine Zeitfenster** von je 23 Millisekunden zerschnitten, und für jedes Fenster fragt eine **Fourier-Transformation** (FFT): *Welche Tonhöhen stecken in diesem Moment drin, und wie laut?* Die Antwort sind 232 Zahlen — eine pro Frequenzband, von tief bis hoch.

43 Zeitschritte × 232 Frequenzen: Das ist eine Tabelle aus Zahlen — und eine Tabelle aus Zahlen ist genau das, was du aus „Bilder als Tensoren“ kennst. **Ein Spektrogramm ist ein Bild.** Waagerecht die Zeit, senkrecht die Tonhöhe, die Helligkeit ist die Lautstärke. Und deshalb steckt hinter der Spracherkennung hier dasselbe **CNN** — dasselbe Faltungs-Prinzip mit Filtern, die Kanten und Muster suchen — wie bei den handgeschriebenen Ziffern in „Ziffern erkennen“. Nur ist das Bild diesmal ein Klang.

```
// Die Reise von 1 Sekunde Zuruf bis zur Antwort — die Shapes
Mikrofon        [44100]       // Luftdruck, 44.100-mal pro Sekunde gemessen
     ↓  in 43 Fenster à 23 ms schneiden, je eine FFT
Spektrogramm    [43, 232, 1]  // 43 Zeitschritte × 232 Frequenzen × 1 Kanal = ein BILD
     ↓  4 Faltungsschichten + Pooling  (dasselbe Prinzip wie in Kapitel 420)
Merkmale        [2000]        // was das Netz aus dem Bild herausgelesen hat
     ↓  eine letzte Dense-Schicht + Softmax
Scores          [20]          // 20 Wahrscheinlichkeiten, zusammen genau 1,0
```

Hier siehst du dieses Bild gleich live. Sobald du unten das Mikrofon startest, malt der Browser das Spektrogramm des zuletzt erkannten Wortes in dieses Feld — links der Anfang der Sekunde, rechts das Ende, unten die tiefen Töne, oben die hohen. Hell = laut.



Noch leer — starte unten das Mikrofon und sag ein Wort. ↓

Interaktives Modul – am Gerät ausprobieren

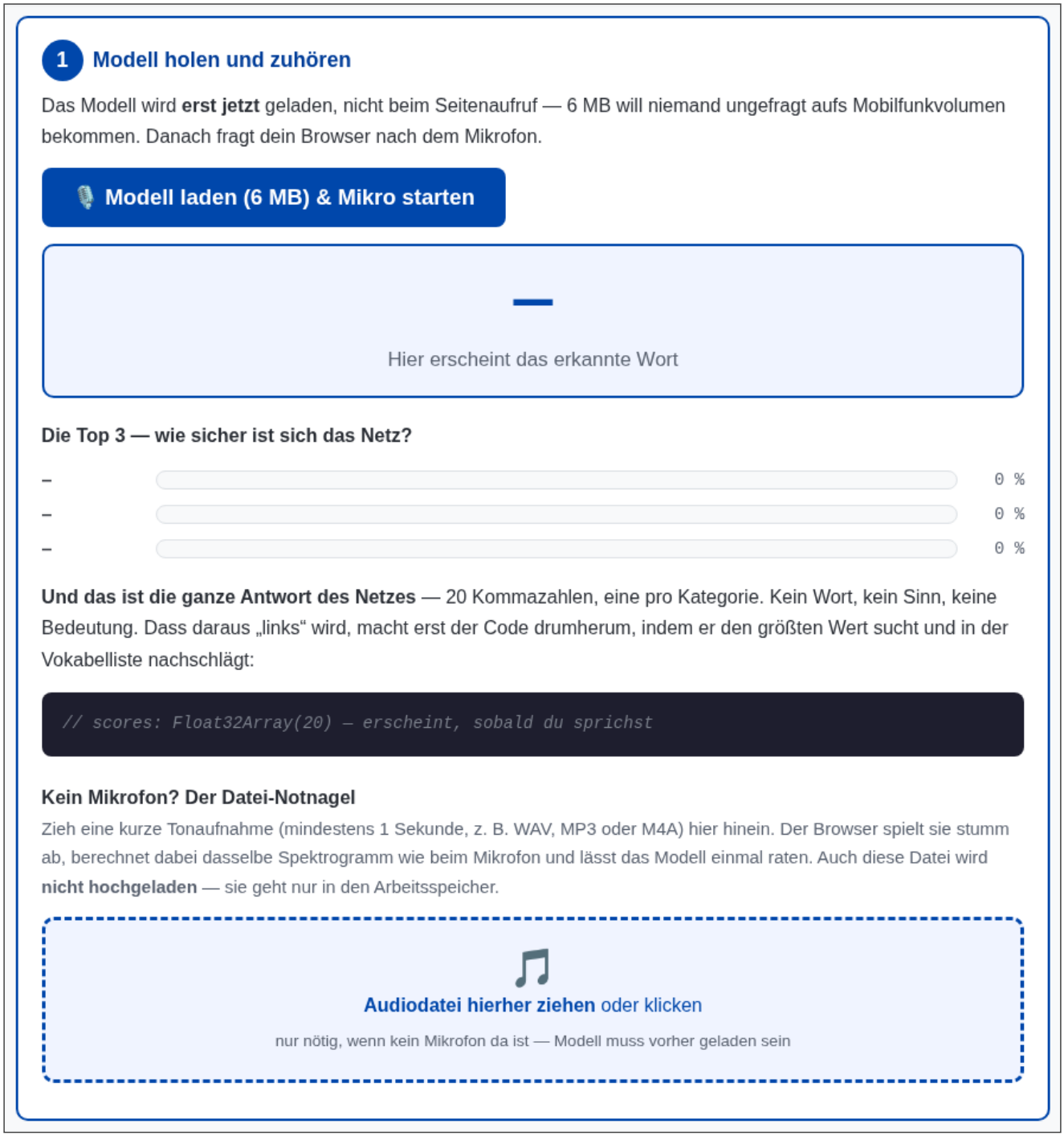
0 ms→ Zeit →1000 ms

Noch leer — starte unten das Mikrofon und sag ein Wort. ↓

Werkzeug 1: Kommandos erkennen

Am Gerät ausprobieren: herrki.de/450_audio_kommandos#abschnitt-5

Direkt zum Modul



Interaktives Modul – am Gerät ausprobieren

Werkzeug 1 in Aktion: Steuer die Figur mit deiner Stimme

Am Gerät ausprobieren: herrki.de/450_audio_kommandos#abschnitt-6

Direkt zum Modul

Jetzt hängen wir an die 20 Zahlen etwas Nützliches: `left`, `right`, `up` und `down` bewegen die blaue Figur um ein Feld, `go` schickt sie in den Dauerlauf in die zuletzt genannte Richtung, `stop` hält sie an. Bring sie auf das rote Ziel — dann springt das Ziel weiter.



Ziele erreicht: **0** Schritte: **0** Status: **bereit**

Ersatz-Steuerung zum Ausprobieren (oder falls kein Mikrofon da ist):

▲ up | ▼ down | ◀ left | ▶ right

▶▶ go | ■ stop

Warum zappelt die Figur nicht? Weil zwei Stellschrauben das verhindern. Das Netz gibt nur dann ein Kommando weiter, wenn es sich zu mindestens **85 %** sicher ist (`probabilityThreshold 0.85`) — sonst wäre jedes Husten ein Zug. Und es schaut sich das 1-Sekunden-Fenster nur alle halbe Sekunde neu an (`overlapFactor 0.5`), statt 43-mal pro Sekunde. Beides zusammen ist die Drossel: lieber ein Kommando zu wenig als zehn zu viel.

Interaktives Modul – am Gerät ausprobieren

Werkzeug 2: Bring dem Modell DEIN Wort bei

Am Gerät ausprobieren: herrki.de/450_audio_kommandos#abschnitt-7

Direkt zum Modul

Bis hier hast du benutzt, was Google trainiert hat. Jetzt drehen wir den Spieß um. Du willst, dass der Browser auf „HerrKI“ hört? Oder auf deinen Namen, „Kaffee“ oder ein Pfeifen? Kein Problem — und du brauchst dafür **nicht** die 105.000 Aufnahmen, mit denen das Originalmodell trainiert wurde. Acht Aufnahmen reichen. Der Trick heißt **Transfer-Learning**.

Die Idee: Das große Netz hat in den vier Faltungsschichten bereits gelernt, wie menschliche Sprache in einem Spektrogramm *aussieht* — wo Vokale breite Bänder malen, wo ein „t“ einen scharfen senkrechten Strich setzt. Dieses Wissen ist wertvoll und völlig unabhängig davon, *welche* Wörter am Ende herauskommen sollen. Also lassen wir es in Ruhe: Das große Netz wird **eingefroren** und dient nur noch als Ohr, als Merkmals-Extraktor. Wir schneiden bloß die letzte Schicht ab — die mit den 20 Ausgängen — und setzen einen **winzigen neuen Kopf** darauf, der nur zwei Ausgänge hat: „dein Wort“ und „irgendwas anderes“.

```
// Was beim Antrainieren passiert – und was nicht
[43,232,1] → CNN (4 Schichten) → Dense 2000 → Mini-Kopf: Dense 2 → [2]
             └─── EINGEFROREN, lernt nichts ──┘   └ lernt ┘

// Größenordnung: das eingefrorene Ohr hat ~1.500.000 Gewichte,
// der neue Kopf nur 2000 × 2 + 2 = 4002. Deshalb reichen 8 Beispiele.
```

Genau dieses Prinzip steckt hinter dem Finetuning großer Sprachmodelle aus „Finetuning oder Prompting?“: Auch dort bleibt das Milliarden-Parameter-Modell im Kern unangetastet, und nur ein hauchdünner Zusatz wird gelernt (bei Sprachmodellen heißt diese Technik **LoRA**). Der Unterschied zu dem, was du gleich machst, ist nur die Größe — das Prinzip ist buchstäblich dasselbe: *Wissen wiederverwenden statt neu erfinden.*

Dein Wort: HerrKI — kurz, gut aussprechbar, immer gleich betont

HerrKI

0 / 8 Aufnahmen

● Aufnehmen (1 s)

Hintergrund

0 / 8 Aufnahmen

● Aufnehmen (1 s)

Für „Hintergrund“ nimmst du **Stille, Raumgeräusch oder ein völlig anderes Wort** auf. Das ist keine Fleißarbeit, sondern nötig: Ein Netz kann nur unterscheiden, wenn es weiß, wovon. Ein Modell, das nur „HerrKI“ kennt, würde auf alles mit „HerrKI!“ antworten.

🚀 Antrainieren (30 Epochen) ↺ Aufnahmen verwerfen

Interaktives Modul – am Gerät ausprobieren

Ehrlich bleiben: was das hier nicht kann

- **Englisch, und nur diese 18 Wörter.** Das Modell hat nie ein deutsches Wort gehört. Es kann auch keine Sätze, keine Namen, keine Grammatik — es ordnet ein 1-Sekunden-Bild einer von 20 Schubladen zu. Mehr ist es nicht, und mehr behauptet es auch nicht.
- **Akzent und Lärm hauen rein.** Deutsche Sprecher betonen „go“ und „no“ anders als die Amerikaner in den Trainingsdaten — Verwechslungen sind normal. Nebengeräusche, Lüfter, Musik oder ein Zimmer mit Hall verschieben das Spektrogramm sichtbar, und das Netz sieht ja nur das Bild.
- **Das 1-Sekunden-Fenster ist hart.** Wer zu langsam spricht, wird abgeschnitten; wer zu früh anfängt, landet halb außerhalb. Deshalb: kurz, klar, mittig ins Fenster.
- **Dein eigenes Wort ist empfindlich.** Acht Beispiele sind acht Beispiele. Sprich beim Antrainieren *und* beim Testen gleich — gleiche Lautstärke, gleicher Abstand, gleiches Zimmer. Wenn du beim Aufnehmen flüsterst und danach rufst, ist das für das Netz ein anderes Bild. Genau deshalb funktionieren Sprachassistenten in der Küche schlechter als im Büro.

Und damit zurück nach oben: Wenn deine Sprachassistentin dich das nächste Mal falsch versteht, weißt du jetzt, was schiefging — irgendwo lag ein Bild auf dem Tisch, das dem falschen Muster ähnlicher sah. Kein Missverständnis, keine schlechte Laune, keine Magie. Nur eine Zahlenreihe mit 20 Einträgen, deren größter an der falschen Stelle stand.

Zurück zur Anwendung

Zurück zu „Alexa, mach das Licht an“: Was in der Sekunde zwischen Zuruf und Reaktion passiert, hast du jetzt selbst laufen sehen — eine Sekunde Ton wird zum Spektrogramm, das Spektrogramm zu einem Bild, das Bild zu 20 nüchternen Zahlen. Du hast damit eine Spielfigur gesteuert und dem Modell mit acht Aufnahmen ein Wort beigebracht, das es vorher nicht kannte. Damit kannst du jetzt auch einschätzen, warum Sprachsteuerung in der lauten Küche schlechter arbeitet als am Schreibtisch — und was ein „falsch verstanden“ in Wahrheit ist.

Der Trick, mit dem dein eigenes Wort so schnell saß, hat einen Namen: **Transfer-Learning**. Im nächsten Kapitel „Deine eigene Bild-KI (Transfer Learning)“ nimmst du ihn auseinander — diesmal mit Bildern statt mit Klang.

Zum Weiterlesen

- Warden 2018, *„Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition“* — arxiv.org/abs/1804.03209 — die Beschreibung genau des Wortschatzes, den dein Browser hier gerade erkannt hat, samt Aufnahme-Bedingungen.
- TensorFlow.js, Modell *speech-commands* — github.com/tensorflow/tfjs-models (github.com/tensorflow/tfjs-models/tree/master/speech-commands) — das Modell aus diesem Kapitel mit der Schnittstelle fürs Nachtrainieren eigener Wörter.
- Google, *Teachable Machine* — teachablemachine.withgoogle.com — dasselbe Transfer-Prinzip als fertiges Werkzeug, wenn du mehr eigene Wörter anlernen willst.

Kurzer Check – mit Lösung

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

Frage 1: Warum sagt man, Spracherkennung sei hier in Wahrheit Bild-Erkennung?

- Weil das Modell nebenbei die Lippen über die Kamera abliest.
- **✓ Weil aus 1 Sekunde Ton ein Spektrogramm wird — eine Tabelle [43, 232, 1] aus Zeit × Frequenz — und die wird von einem CNN wie ein Bild verarbeitet.**
- Weil die Wörter vorher als Text-Bilder gerendert und dann verglichen werden.

Waagerecht die Zeit, senkrecht die Tonhöhe, Helligkeit = Lautstärke. Für das Netz ist das exakt dieselbe Sorte Tensor wie eine handgeschriebene Ziffer in Kapitel 420 — und deshalb funktioniert dort wie hier dasselbe Faltungs-Prinzip.

Frage 2: Du hast dem Modell dein eigenes Wort beigebracht. Was wurde dabei tatsächlich trainiert?

- Das komplette Netz mit allen ~1,5 Millionen Gewichten wurde neu berechnet.
- **✓ Nur ein winziger neuer Kopf mit rund 4.000 Gewichten — die vier Faltungsschichten blieben eingefroren und dienten als Ohr.**
- Gar nichts — deine Aufnahmen wurden nur gespeichert und beim Erkennen direkt verglichen.

Das ist der ganze Transfer-Learning-Trick: Das teuer trainierte Wissen „wie sieht Sprache im Spektrogramm aus“ wird wiederverwendet, gelernt wird nur die letzte Zuordnung. Deshalb reichen 8 Beispiele und ein paar Sekunden Rechenzeit statt 105.000 Aufnahmen und Tagen im Rechenzentrum — genau wie LoRA bei Sprachmodellen (Kapitel 370).

Frage 3: Warum bleibt dein Mikrofon-Ton hier privat, anders als bei Alexa oder Siri?

- Weil der Server die Aufnahmen sofort nach der Erkennung wieder löscht.
- Weil das Audio vor dem Hochladen verschlüsselt wird.
- **✓ Weil gar nichts hochgeladen wird: Das 6-MB-Modell kommt zu dir, und Spektrogramm, Erkennung und Training laufen komplett in deinem Browser.**

Die Richtung ist umgedreht — nicht die Daten reisen zum Modell, sondern das Modell zu den Daten. Bei den großen Sprachassistenten geht der Ton (oder eine daraus berechnete Darstellung) ins Rechenzentrum, weil deren Modelle viel zu groß fürs Gerät sind. Genau dieser Kompromiss ist Thema in „Lokale KI“.

Das nimmst du mit

- **Hören ist hier Bild-Erkennung.** Der Browser malt deinen Ton — und ein Bilderkennungs-Netz schaut sich das Bild an.
- **Ein Spektrogramm ist ein Tensor `[43, 232, 1]`.** 43 Zeitfenster von je 23 Millisekunden, pro Fenster 232 Frequenzbänder aus der Fourier-Transformation.
- **Am Ende stehen 20 Zahlen.** Wahrscheinlichkeiten, zusammen genau 1,0 — mehr sagt das Netz nicht, alles Weitere ist unsere Zuordnung zu `left`, `right` und Co.
- **Transfer-Learning heißt: einfrieren und einen winzigen Kopf antrainieren.** Genau so hast du dem fertigen Modell dein eigenes Wort beigebracht — in Sekunden statt Stunden.
- **Nicht deine Daten reisen, sondern das Modell.** 6 MB kommen zu dir; Spektrogramm, Erkennung und Training laufen komplett in deinem Browser.

Weiter geht’s: Das CNN, das hier deine Klang-Bilder liest, hast du in „Ziffern erkennen“ selbst trainiert — dort auf Ziffern, hier auf Spektrogrammen, identisches Prinzip. Und das Einfrieren-plus-Mini-Kopf-Verfahren von eben ist im Kleinen genau das, was in „Finetuning oder Prompting?“ mit Milliarden-Modellen gemacht wird. Im nächsten Kapitel wendest du dasselbe Transfer-Prinzip auf **Bilder** an und baust dir mit ein paar Webcam-Schnappschüssen deine eigene Bild-KI. Gleicher Trick, anderes Sinnesorgan.

So zitierst du dieses Kapitel

HerrKI: Sprich mit deinem Browser (Audio). herrki.de/450_audio_kommandos, ein privates Hobbyprojekt von Terragon.de, abgerufen am 09.09.2026.