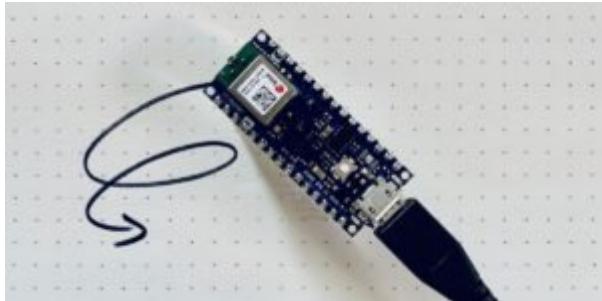


# Komplexe Gesten erkennen mit künstlicher Intelligenz und deinem Arduino



In diesem Projekt entwickelst du ein KI-Modell, das Gesten wie Kreisbewegungen und Winken erkennen kann. Anschließend lädst du es auf deinen Arduino Nano 33 BLE Sense und lässt dir die erkannten Bewegungen im Terminal deines PCs oder Macs anzeigen.

Mit diesem Setting kannst du später aufwändigeres Projekte bauen, um z.B. das Licht durch Gesten an- und ausschalten zu können.

**Für dieses Projekt benötigst du:**

Einen kostenlosen Account bei Edge Impulse und einen Arduino Nano 33 BLE Sense.



## [Arduino® Board Nano 33 BLE Sense with headers](#)

Arduino Board Nano 33 BLE Sense with headers; Inhalt: 1 Stück

## Vorbereitungen

In unserem Tutorial [Gesten erkennen mit dem Sensor APDS9960](#) haben wir beschrieben, wie du mit dem Arduino Nano 33 BLE Sense Bewegungen nach oben, unten, rechts und links erkennen kannst. **Dieser Microcontroller kann aber deutlich mehr!**

Du kannst ihn mit ein paar Vorbereitungen mit dem Service von Edge Impulse verbinden und seinen Beschleunigungssensor verwenden, um viel anspruchsvollere Gesten und Muster erkennen zu können.

Lies zunächst nach, wie du deinen [Arduino mit Edge Impulse verbindest](#).

## Bewegungsdaten sammeln

Am Anfang jeder künstlichen Intelligenz stehen Daten. Das bedeutet, dass du zunächst Daten zu den Bewegungen sammeln musst, die du später voneinander unterscheiden möchtest.

In diesem Tutorial lernst du, wie du [Bewegungsdaten mit dem Arduino Nano 33 BLE Sense sammeln und speichern kannst](#). Auf dieses Tutorial bauen wir hier auf.

In diesem Projekt nutzt du dieses Wissen und bringst deinem Arduino bei zu erkennen, ob du eine Kreisbewegung ausführst, in der Luft winkst oder – nichts tust. Öffne in Edge Impulse als den Menüpunkt **Data acquisition** und sammle Samples für die zwei Bewegungen und den Ruhezustand – jeweils circa 3 Minuten.

Achte darauf, dass sich deine Bewegungen innerhalb einer Geste nicht zu sehr unterscheiden. Eine leichte Varianz ist aber normal und auch gut. Wenn du fertig bist, sollte dein Screen ungefähr so aussehen:

SAMPLE NAME	LABEL	ADDED	LENGTH	⋮
circle.1gi9v0vk	circle	Today, 17:14...	10s	⋮
circle.1gi9ui01	circle	Today, 17:13...	10s	⋮
circle.1gi9u2oi	circle	Today, 17:13...	10s	⋮
circle.1gi9tkg5	circle	Today, 17:13...	10s	⋮
circle.1gi9t61a	circle	Today, 17:13...	10s	⋮

Anschließend geht es mit dem Training des KI-Modells weiter.

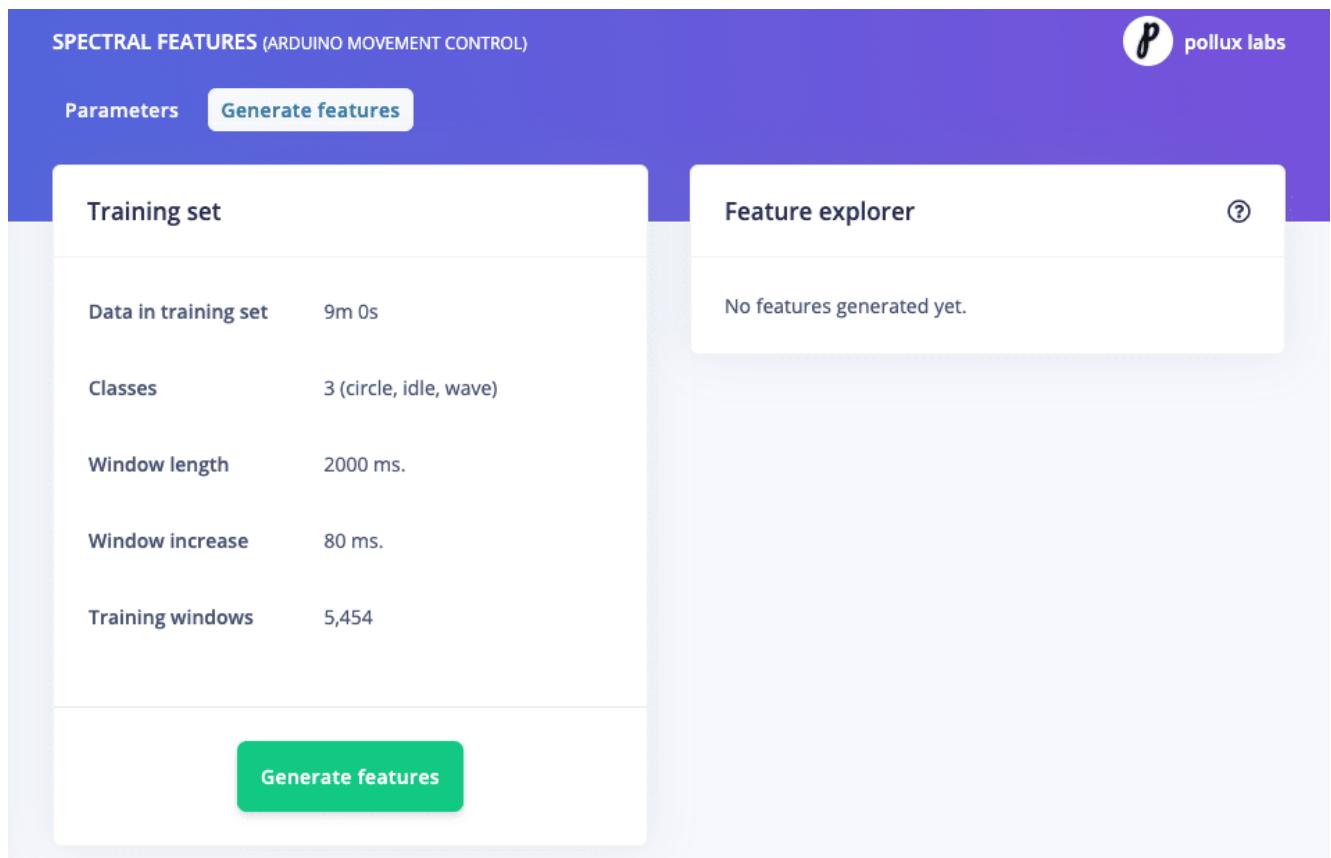
## Einen Impulse entwickeln

Jetzt geht es weiter mit dem nächsten Menüpunkt: **Impulse design**. Hier entwickelst du aus deinen Rohdaten sogenannte Features, mit denen später deine Bewegungen analysiert und zugeordnet werden können.

Links siehst du die Karte **Time series data** – hier musst du zunächst nichts einstellen. Klicke stattdessen rechts daneben auf **Add a processing block** und wähle anschließend den Eintrag **Spectral Analysis**. Dieser Block eignet sich besonders für Bewegungsdaten.

Wähle eine Karte weiter den Eintrag **Neural Network (Keras)**. Klicke zuletzt auf **Save Impulse**.

Neben dem Menüpunkt **Create impulse** sollte nun ein grüner Punkt erscheinen. Klicke nun gleich darunter auf **Spectral features** und anschließend oben auf **Generate features**.



Hier gibt es nichts weiter zu tun als auf den Button **Generate features** zu klicken. Diese Prozedur läuft nun durch und sollte nur wenige Sekunden dauern. Sobald sie abgeschlossen ist, siehst du rechts eine Visualisierung der Features und der Cluster, die sie bilden.

## Feature explorer (5.454 samples)



X Axis

Y Axis

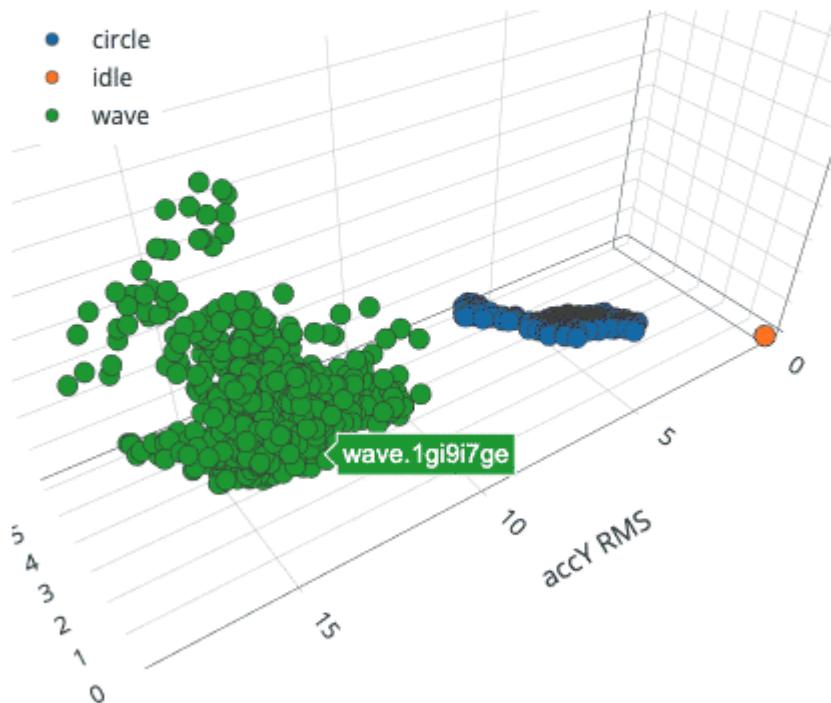
Z Axis

accX RMS

accY RMS

accZ RMS

- circle
- idle
- wave



Wie du oben siehst, bilden die verschiedenen farbigen Punkte eigene Cluster, die voneinander entfernt liegen. Das sollte auch bei deinen Bewegungsdaten der Fall sein. **Je besser du die Cluster voneinander unterscheiden kannst, desto besser kann das auch später dein KI-Modell.**

Solltest du nur einen Haufen bunter Punkte sehen, gehe noch einmal zurück und sammle weitere oder neue Bewegungsdaten.

## Das neuronale Netz trainieren

Edge Impulse hat nun klar definierte Daten und Klassen, die

gut voneinander zu unterscheiden sind. Fehlt nur noch ein neuronales Netz, das neue Daten einem dieser Klassen zuordnen kann.

Kurz gesagt, ist ein neuronales Netz nichts anderes als Algorithmen, die versuchen Muster zu erkennen. [Auf Wikipedia lernst du mehr über dieses Thema.](#)

Klicke nun im Menü auf den nächsten Punkt **NN Classifier**.

Neural Network settings

Training settings

Number of training cycles ② 100

Learning rate ② 0.0001

Minimum confidence rating ② 0.80

Neural network architecture

Input layer (33 features)

Dense layer (20 neurons)

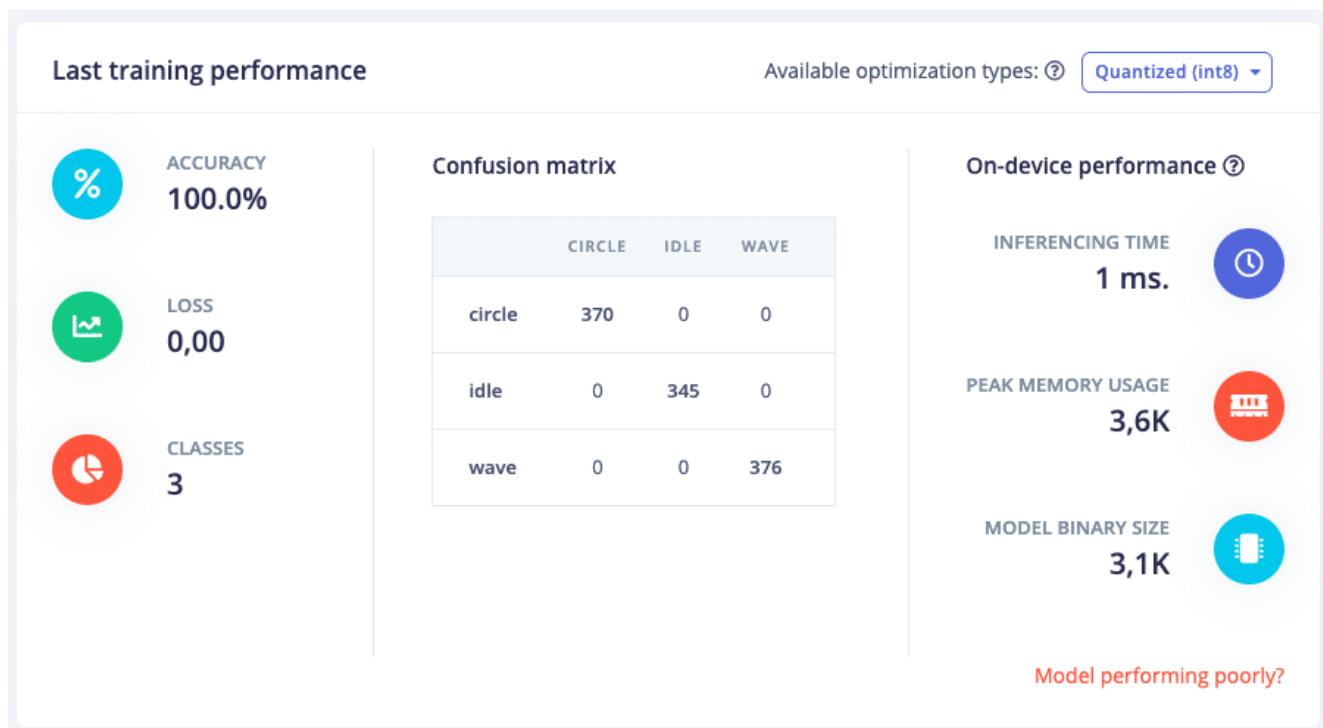
Dense layer (10 neurons)

Add an extra layer

Output layer (3 features)

Belasse die Einstellungen zunächst so wie sie sind und klicke auf den Button **Start training**. Das nun folgende Training des Netzes sollte wenige Minuten in Anspruch nehmen.

Anschließend erscheint eine neue Karte mit den Ergebnissen. Oben links siehst du unter **Accuracy**, wie akkurat dein neuronales Netz arbeitet. In unserem Fall kam eine Genauigkeit von 100% heraus, was darauf hoffen lässt, dass die Bewegungen und Gesten richtig erkannt werden.



Wenn du eine niedrigere Genauigkeit erhalten hast, findest du hinter dem Link **Model performing poorly?** unten rechts Tipps, wie du ein besseres KI-Modell entwickeln kannst.

## Ein erster Test der künstlichen Intelligenz

Jetzt wird es spannend: Erkennt deine KI die Bewegungen, die du mit deinem Arduino Nano 33 BLE Sense ausführst, richtig?

Klicke im Menü auf den Punkt **Live classification**. Hier kannst einen ersten Test starten, indem du auf den Button **Start**

**sampling** klickst und eine der trainierten Bewegungen ausführst.

Nachdem du das Sample aufgenommen hast, erscheint das Ergebnis. Wir haben eine Kreisbewegung ausgeführt, was die künstliche Intelligenz richtig der Klasse **circle** zugeordnet hat.

Classification result	
Summary	
Name	testing.1gic6s6a
Expected outcome	testing
Category	
CATEGORY	COUNT
circle	101
idle	0
wave	0
uncertain	0

Sollte das bei dir nicht funktioniert haben, gehe am besten wieder ein paar Schritte zurück und versuche bessere Bewegungsdaten zu sammeln oder dein neuronales Netz besser zu trainieren.

Wenn du aber zufrieden bist, wird es Zeit die KI auf deinen Arduino zu bringen.

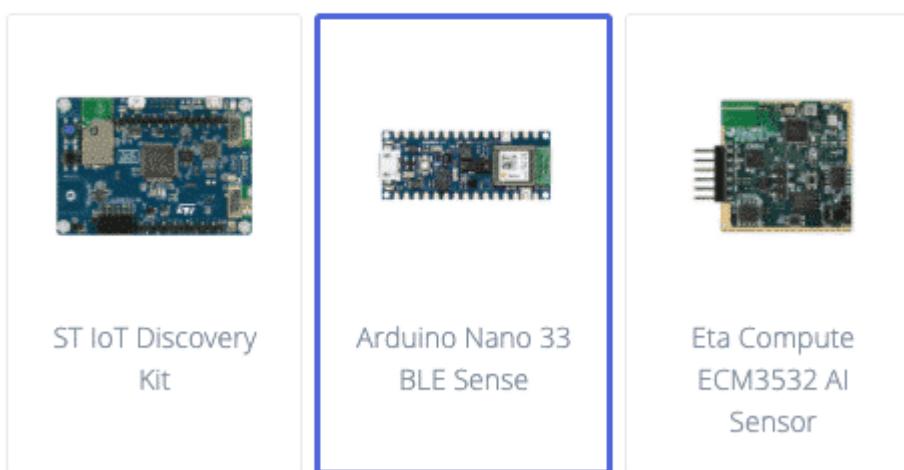
# Die KI auf dem Arduino ausführen

Klicke hierfür auf den Menüpunkt **Deployment** und wähle unter der Überschrift **Build firmware** den Arduino Nano 33 BLE Sense.

Du kannst dir auch eine Arduino-Bibliothek erstellen lassen, die du in einen Sketch einbinden kannst. Wir bleiben hier jedoch bei der einfacheren Variante.

## Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.



Weiter unten hast du auch die Möglichkeit, Optimierungen an deinem KI-Modell vorzunehmen, aber auch diese lassen wir hier erst einmal beiseite.

Klicke als nächstes auf den Button **Build**. Jetzt wird die Firmware mit deinem KI-Modell erstellt, was wieder einige Minuten dauern kann.

Als nächstes erhältst du eine Zip-Datei mit der Firmware. Führe, je nachdem welches Betriebssystem du verwendest, eine der folgenden Dateien aus:

- `flash_windows.bat`
- `flash-mac.command`
- `flash_linux.sh`

Achte darauf, dass du das Terminal, in dem das Sampling der Daten läuft, vorher schließt und diesen Prozess beendest.

Öffne nach dem Installieren der Firmware ein neues Terminal-Fenster und führe diesen Befehl aus:

```
edge-impulse-run-impulse
```

Jetzt läuft deine KI in diesem Terminal und prüft alle zwei Sekunden, ob sie eine Bewegung deines Arduinos erkennt und wenn ja, welche. Hier hat sie mit einer Wahrscheinlichkeit von 96% ein Winken erkannt – was stimmte:

```
Predictions (DSP: 22 ms., Classification: 1 ms., Anomaly: 0 ms.):
  circle: 0.03516
  idle: 0.00000
  wave: 0.96484
Starting inferencing in 2 seconds...
Sampling...
```

## Wie geht es weiter?

Du hast jetzt eine funktionierende künstliche Intelligenz auf deinem Arduino, die komplexe Bewegungen und Gesten erkennen kann.

Gehe einen Schritt weiter und integriere sie in ein neues Projekt, das du mit Gesten steuern kannst. Dein Arduino Nano 33 BLE Sense verfügt auch über ein Mikrofon, das du in Edge Impulse verwenden kannst. Sammle Geräusche und entwickle eine KI, die diese unterscheiden kann. Deiner Fantasie sind keine Grenzen gesetzt! ☐

Edge Impulse ist übrigens nicht der einzige Service, mit dem relativ leicht KI-Modelle erstellen kannst: [In diesem Tutorial stellen wir dir Googles Teachable Machine vor.](#)