

# Custom Wake-Word Training Guide

---

## Übersicht

---

Dieses Projekt nutzt ein custom-trainiertes “**Computer**” Wake-Word mit [Porcupine Wake Word Detection](#) von Picovoice. Das Wake-Word ermöglicht eine hands-free Aktivierung des Voice Assistants durch einfaches Aussprechen von “Computer” .

## Warum “Computer” ?

---

Die Wahl des Wake-Words “Computer” basiert auf mehreren Überlegungen:

**Inspiration:** Das Wake-Word ist inspiriert von Star Trek, wo die Crew den Bordcomputer mit “Computer” aktiviert. Dies schafft eine vertraute und futuristische Nutzererfahrung.

**Phonetische Eigenschaften:** “Computer” ist ein ideales Wake-Word, da es mehrere wichtige Kriterien erfüllt. Mit drei Silben (Com-pu-ter) ist es lang genug, um eindeutig erkennbar zu sein, aber kurz genug, um natürlich ausgesprochen zu werden. Die Kombination aus harten Konsonanten (C, P, T) und weichen Vokalen (o, u, e) macht es akustisch gut unterscheidbar. Das Wort ist in der deutschen Sprache eindeutig und wird selten in normalen Gesprächen verwendet, was die Anzahl von Falsch-Positiven reduziert.

**Praktische Vorteile:** Im Vergleich zum vorherigen Wake-Word “hey jarvis” bietet “Computer” eine kürzere Aktivierungsphrase, was die Nutzung schneller und effizienter macht. Zudem ist es sprachübergreifend verständlich und kann später auch in anderen Sprachen genutzt werden.

# Training-Methode: Porcupine

---

## Warum Porcupine?

Porcupine wurde als Wake-Word-Engine gewählt, weil es mehrere entscheidende Vorteile bietet:

**Transfer Learning:** Porcupine nutzt vortrainierte akustische Modelle, die bereits auf großen Mengen von Sprachdaten trainiert wurden. Beim Training eines neuen Wake-Words wird lediglich der Phrasentext eingegeben, und das System adaptiert das Basismodell automatisch. Dies eliminiert die Notwendigkeit, hunderte von Aufnahmen zu sammeln.

**Geschwindigkeit:** Der gesamte Training-Prozess dauert nur wenige Sekunden, verglichen mit mehreren Stunden bei traditionellen Methoden. Dies ermöglicht schnelles Prototyping und Testing verschiedener Wake-Words.

**Qualität:** Trotz des vereinfachten Prozesses liefert Porcupine produktionsreife Modelle mit hoher Genauigkeit. Die Engine wird von Unternehmen wie NASA eingesetzt, was die Zuverlässigkeit unterstreicht.

**Plattform-Unterstützung:** Porcupine unterstützt Windows, Linux, macOS, iOS, Android, Web und Embedded-Systeme. Dies ermöglicht die zukünftige Expansion des Voice Assistants auf andere Geräte.

**Kosteneffizienz:** Für persönliche Projekte und Entwicklung ist Porcupine kostenlos nutzbar (Free Tier), was es ideal für Open-Source-Projekte macht.

## Alternative: OpenWakeWord

Als Open-Source-Alternative steht OpenWakeWord zur Verfügung. Diese Methode erfordert mehr Aufwand (Aufnahme von 200-500 Samples, Training via Google Colab), bietet aber vollständige Kontrolle über das Modell und keine Abhängigkeit von kommerziellen Services. Eine detaillierte Anleitung für OpenWakeWord-Training ist in `02_computer_training_guide.md` verfügbar.

# Training-Prozess (Porcupine)

---

## Schritt 1: Picovoice Account erstellen

1. Navigiere zu [Picovoice Console](#)
2. Registriere dich mit deiner E-Mail-Adresse
3. Bestätige die E-Mail und logge dich ein
4. Kopiere deinen **AccessKey** aus dem Account-Bereich

**Wichtig:** Der AccessKey ist vertraulich und sollte nicht öffentlich geteilt werden. Speichere ihn in einer `.env` Datei (siehe unten).

## Schritt 2: Wake-Word trainieren

1. Öffne die [Porcupine-Seite](#) in der Console
2. Wähle **English** als Sprache
3. Gib **“Computer”** als Wake-Word ein
4. (Optional) Teste das Wake-Word mit dem Mikrofon-Button
5. Klicke auf **“Train”** und warte 5-10 Sekunden
6. Wähle **Windows** als Plattform
7. Klicke auf **“Download”** und speichere die `.ppn` Datei

## Schritt 3: Modell integrieren

1. Erstelle einen `models` Ordner im Projektverzeichnis:

```
mkdir models
```

2. Verschiebe die heruntergeladene `.ppn` Datei in den `models` Ordner:

```
move Downloads\computer_windows.ppn models\computer.ppn
```

3. Erstelle eine `.env` Datei im Projektverzeichnis:

```
PICOVOICE_ACCESS_KEY=dein_access_key_hier
```

4. Installiere die erforderlichen Pakete:

```
pip install pvporcupine python-dotenv
```

5. Aktualisiere `.gitignore`: “`

## Environment variables

---

`.env`

# Porcupine models \*.ppn models/

```

## Verwendung

### Code-Integration

Der Voice Assistant nutzt Porcupine wie folgt:

```python
import pvporcupine
from dotenv import load_dotenv
import os

# Lade AccessKey
load_dotenv()
ACCESS_KEY = os.getenv('PICOVOICE_ACCESS_KEY')

# Initialisiere Porcupine
porcupine = pvporcupine.create(
    access_key=ACCESS_KEY,
    keyword_paths=['models/computer.ppn'],
    sensitivities=[0.5] # 0.0-1.0
)

# Verarbeite Audio
keyword_index = porcupine.process(audio_frame)
if keyword_index >= 0:
    print("Computer wake word detected!")

```

## Sensitivity-Anpassung

Die Sensitivity steuert die Balance zwischen Erkennungsrate und Falsch-Positiven:

- **0.3:** Sehr konservativ, wenige Falsch-Positive, aber möglicherweise niedrigere Erkennungsrate
- **0.5:** Balanced (Standard), gute Balance zwischen Erkennung und Falsch-Positiven
- **0.7:** Aggressiv, hohe Erkennungsrate, aber mehr Falsch-Positive möglich

Passe die Sensitivity basierend auf deiner Umgebung an:

```
porcupine = pvporcupine.create(  
    access_key=ACCESS_KEY,  
    keyword_paths=['models/computer.ppn'],  
    sensitivities=[0.7] # Für lautere Umgebungen  
)
```

## Eigenes Wake-Word trainieren

---

### Schnellstart mit Porcupine

Wenn du ein anderes Wake-Word verwenden möchtest (z.B. “Jarvis” , “Alexa” , “Hey Assistant”):

1. Folge dem Training-Prozess oben, aber gib dein gewünschtes Wake-Word ein
2. Beachte die [Wake-Word-Auswahl-Richtlinien](#):
  - 2-4 Silben
  - Mix aus Vokalen und Konsonanten
  - Eindeutige Phoneme
  - Leicht auszusprechen
3. Lade das neue Modell herunter und ersetze `computer.ppn`
4. Aktualisiere `WAKE_WORD` im Code

### Detaillierte Anleitung mit OpenWakeWord

Für vollständige Kontrolle und Open-Source-Lösung siehe die detaillierte Anleitung in [02\\_computer\\_training\\_guide.md](#) . Diese Methode erfordert:

- Aufnahme von 200-500 Wake-Word-Samples
- Sammlung von negativen Samples
- Training via Google Colab (2-4 Stunden)
- Mehr technisches Verständnis

Ein automatisiertes Recording-Skript ist verfügbar: [03\\_record\\_wake\\_word.py](#)

# Performance-Metriken

---

## Erwartete Werte

Für ein gut trainiertes “Computer” Wake-Word sollten folgende Metriken erreicht werden:

**Erkennungsrate:** >95% bei normaler Aussprache in ruhiger Umgebung **Falsch-Positiv-Rate:** % pro Stunde normaler Konversation **Latenz:** <50ms von Wort-Ende bis Erkennung **CPU-Auslastung:** % auf modernen Systemen **Memory-Footprint:** <50MB

## Testing

Teste das Wake-Word systematisch:

```
# Basis-Test
python test_porcupine.py

# Umfangreiche Tests
python 09_test_wake_word.py
```

Siehe [08\\_wake\\_word\\_testing.md](#) für detaillierte Test-Checklisten.

## Troubleshooting

---

### Problem: Wake-Word wird nicht erkannt

#### Lösungen:

1. Erhöhe Sensitivity: `sensitivities=[0.7]`
2. Überprüfe Mikrofon-Lautstärke in Windows
3. Sprich deutlicher und näher am Mikrofon
4. Reduziere Hintergrundgeräusche

## Problem: Zu viele Falsch-Positive

### Lösungen:

1. Senke Sensitivity: `sensitivities=[0.3]`
2. Implementiere Cooldown-Period (bereits im Code)
3. Teste in verschiedenen Umgebungen

## Problem: “Invalid AccessKey”

### Lösungen:

1. Überprüfe `.env` Datei: Kein Leerzeichen, keine Anführungszeichen
2. Kopiere AccessKey erneut von [Picovoice Console](#)
3. Stelle sicher, dass `.env` im Projektverzeichnis liegt

Weitere Probleme und Lösungen: [10\\_troubleshooting.md](#)

## Ressourcen

---

### Dokumentation

- [Porcupine Wake Word Docs](#)
- [Porcupine Python SDK](#)
- [Wake-Word-Auswahl-Guide](#)

### Alternative Methoden

- [OpenWakeWord GitHub](#)
- [Mycroft Precise](#)

### Tools

- [Recording-Skript](#) - Automatisierte Aufnahmen für OpenWakeWord
- [Test-Skript](#) - Systematisches Wake-Word-Testing



- [Training-Guide](#) - Detaillierte Schritt-für-Schritt-Anleitung

## Lizenz

---

Das “Computer” Wake-Word-Modell wurde mit Porcupine trainiert und unterliegt den [Picovoice Nutzungsbedingungen](#). Die Free Tier ist für persönliche Projekte und Entwicklung kostenlos nutzbar.

Der Voice Assistant Code ist unter der MIT-Lizenz verfügbar (siehe [LICENSE](#)).

## Beitragen

---

Verbesserungen und Feedback zum Wake-Word-Training sind willkommen! Bitte öffne ein Issue oder Pull Request auf GitHub.

---

**Erstellt:** 05. Dezember 2025

**Projekt:** Computer Voice Assistant

**GitHub:** [https://github.com/KoMMb0t/voice\\_assi](https://github.com/KoMMb0t/voice_assi)