

Computer Voice Assistant - Overnight Work Deliverables

Datum: 05. Dezember 2025

Projekt: Computer Voice Assistant

Hauptprojekt: [voice_assi](#)

Übersicht

Dieses Repository enthält alle **Overnight Work Deliverables** für das Computer Voice Assistant Projekt. Insgesamt wurden **10 parallele Aufgaben** vorbereitet, um das Wake-Word-Training und die LLM-Integration zu planen.



Deliverables

1. Wake-Word-Methoden-Vergleich

Datei: `01_wake_word_comparison.md`

Inhalt: Detaillierter Vergleich von Porcupine, OpenWakeWord und Snowboy mit Vor-/Nachteilen, Kosten und Empfehlungen.

2. Computer Wake-Word Trainings-Anleitung

Datei: `02_computer_training_guide.md`

Inhalt: Schritt-für-Schritt-Anleitung für beide Methoden (Porcupine: 5 Min, OpenWakeWord: 4-8 Std).

3. Recording-Skript

Datei: 03_record_wake_word.py

Inhalt: Automatisiertes Python-Skript für Aufnahmen (200 positive + 200 negative + 60 background Samples).

4. Code-Integration

Datei: 04_voice_assistant_computer.py

Inhalt: Vollständiger Voice Assistant mit Porcupine “Computer” Wake-Word, Cooldown-System, alle Befehle.

5. GitHub-Dokumentation

Dateien:

- 05_WAKE_WORD_TRAINING.md - Haupt-Dokumentation
- 06_README_UPDATE.md - README Änderungen
- 07_GITIGNORE_UPDATE.txt - .gitignore Ergänzungen

6. Testing-Framework

Datei: 08_wake_word_testing.md

Inhalt: Umfassende Test-Checklisten (Erkennungsrate, Falsch-Positive, Umgebung, Stress, Performance).

7. Troubleshooting-Guide

Datei: 10_troubleshooting.md

Inhalt: Lösungen für 8 Problemkategorien mit Code-Beispielen.

8. Assets-Sammlung

Datei: 11_assets_collection.md + assets/ Ordner

Inhalt: 8 Icons/Logos + Dokumentation für Branding.

9. LLM-Integration Architektur

Datei: 12_llm_architecture.md

Inhalt: Vollständige Architektur-Planung für ChatGPT, Perplexity, Manus-Integration.

10. Roadmap & Next Steps

Datei: 13_roadmap_next_steps.md

Inhalt: Projekt-Planung (kurzfristig, mittelfristig, langfristig) mit Milestones.



Formate

Alle Deliverables sind in **4 Formaten** verfügbar:

1. **Markdown (.md)** - Original-Format, GitHub-optimiert
 2. **PDF (.pdf)** - Druckbar, professionell formatiert (in `pdf_exports/`)
 3. **CSV (.csv)** - Übersicht aller Deliverables (`DELIVERABLES_OVERVIEW.csv`)
 4. **ZIP (.zip)** - Komplettes Archiv zum Download
-



Quick Start

1. Repository klonen

```
git clone https://github.com/KoMMb0t/Computer-Voice-Assi.git  
cd Computer-Voice-Assi
```

2. Dokumentation lesen

```
# Hauptdokumentation  
cat 05_WAKE_WORD_TRAINING.md  
  
# Roadmap  
cat 13_roadmap_next_steps.md
```

3. Code testen

```
# Recording-Skript  
pip install sounddevice numpy wave  
python 03_record_wake_word.py  
  
# Voice Assistant (benötigt Porcupine AccessKey)  
pip install pyporcupine vosk edge-tts sounddevice pygame webrtcvad python-dotenv  
python 04_voice_assistant_computer.py
```

Verzeichnis-Struktur

```
Computer-Voice-Assi/
├── 01_wake_word_comparison.md
├── 02_computer_training_guide.md
├── 03_record_wake_word.py
├── 04_voice_assistant_computer.py
├── 05_WAKE_WORD_TRAINING.md
├── 06_README_UPDATE.md
├── 07_GITIGNORE_UPDATE.txt
├── 08_wake_word_testing.md
├── 10_troubleshooting.md
├── 11_assets_collection.md
├── 12_llm_architecture.md
├── 13_roadmap_next_steps.md
├── DELIVERABLES_OVERVIEW.csv
├── README.md
└── assets/
    ├── 4gRLVK12tORi.png
    ├── 4w9TtynImUmE.png
    ├── cdEAdwWksumG.jpg
    └── ... (8 Dateien)
└── pdf_exports/
    ├── 01_wake_word_comparison.pdf
    ├── 02_computer_training_guide.pdf
    └── ... (11 PDFs)
```

Nächste Schritte

1. **Sofort:** Integriere Dokumentation ins Hauptprojekt
2. **Diese Woche:** Teste Wake-Word mit `08_wake_word_testing.md`
3. **Nächste Woche:** Starte LLM-Integration gemäß `12_llm_architecture.md`
4. **Langfristig:** Folge Roadmap in `13_roadmap_next_steps.md`



Beitragen

Feedback und Verbesserungsvorschläge sind willkommen!

Hauptprojekt: https://github.com/KoMMb0t/voice_assi



Lizenz

MIT License - Siehe Hauptprojekt

Erstellt: 05. Dezember 2025

Autor: KoMMb0t

Projekt: Computer Voice Assistant