

Morse-Code Erkennung und Interpretation aus Handgesten-Videos

Maschinelles Sehen
Sommersemester 2025
Prof. Dr. Hildebrand



Fachbereich VI,
Studiengang
Technische Informatik

Khaled Rafei, 930677

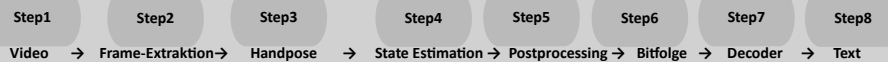
1 EINLEITUNG & MOTIVATION

- Morsecode: robustes Kommunikationssystem
- Idee: Erkennung mit Handgesten (offen = 1, geschlossen = 0)
- Ziel: Video einlesen und die Bitfolgen Dekodierung

Stand der Technik:

- MediaPipe Hands für Landmark-Erkennung
- Heuristik statt ML – ressourcenschonend
- Einschränkung: nur in kontrollierten Szenarien zuverlässig

2 METHODIK & PIPELINE



Implementierung:

- Python, OpenCV, MediaPipe, Numpy

- Module:

video_io.py,
state_estimator.py,
decoder.py

- Parameter:

STABLE_MIN_FRAMES=31,
GAP_TOLERANCE=20

```
MORSE_TABLE_STR = {
    'A': '01', 'B': '1000', 'C': '1010', 'D': '100', 'E': '0',
    'F': '0010', 'G': '110', 'H': '0000', 'I': '00', 'J': '0111',
    'K': '101', 'L': '0100', 'M': '11', 'N': '10', 'O': '111',
    'P': '0110', 'Q': '1101', 'R': '010', 'S': '000', 'T': '1',
    'U': '001', 'V': '0001', 'W': '011', 'X': '1001', 'Y': '1011',
    'Z': '1100', ' ': ''
}

# --- Wörter ---
BASE_WORDS = {'HALLO', 'HELLO', 'WELT', 'TEST', 'HILFE', 'MORSE', 'CODE',
               'ICH', 'MEIN', 'NAME', 'IST', 'BRAUCHE', 'ARSHIA', 'ELHAM',
               'LOVE', 'MOVE', 'HELP', 'SADRA', 'KASRA'}
```

3 EVALUATION & ERGEBNISSE

- Testvideos: 'HELLO. mp4', 'HALLO. mp4', 'HELP. mp4', 'HILFE. mp4', 'LOVE. mp4', 'MOVE. mp4', Namen (Wie: 'ARSHIA. mp4', 'ELHAM.mp4' usw.)
- Testwörter: 'MEINNAME', 'MEINNAMEIST', 'ICHBRAUCHEHILFE' usw.

- Metrik: **Hamming-Distanz**

- Ergebnis: **100 % Genauigkeit bei Lexikon-Wörtern**

- Fehler bei unbekannten Wörtern (z.B. 'INFORMATIK')

Testwörter1: 'INFORMATIK'

erwartete = [0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1]
Ausgabe = [0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1]
Beste Wort-Dekodierung: 'ELHAM WELT' oder 'elham welt'
Distanz: 7 | Genauigkeit: 70.83%

Test1: 'HELLO.mp4'

erwartete = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1]
Ausgabe = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1]
Beste Wort-Dekodierung: 'HELLO' oder 'hello'
Distanz: 0 | Genauigkeit: 100.0%

Test2: 'HALLO.mp4'

erwartete = [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1]
Ausgabe = [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1]
Beste Wort-Dekodierung: 'HALLO' oder 'hallo'
Distanz: 0 | Genauigkeit: 100.0%

Abstand (Cm)	Bis 110	110-150	150-200
Weiß-hell	perfekt	gut	gut
Weiß-halb hell	perfekt	gut	×
Schwarz-hell	perfekt	gut	×
Schwarz-halb hell	perfekt	×	×

4 DISCUSSION

- **Stärken:** hohe Genauigkeit, einfache Pipeline
- **Schwächen:** Lexikonabhängig, sensibel für Beleuchtung
- **Verbesserungen:** ML-Modelle, größeres Lexikon

5 CONCLUSION

- Gesten→Morse→Text funktioniert stabil
- Praxistauglich in einfachen Szenarien
- **Zukunft:** robustere Erkennung, freie Vokabulare

REFERENCES

- Example scenario, M. Shamsi, Le Mans University, 2024
- <https://github.com/google-ai-edge/mediapipe/tree/master/mediapipe>
- <https://www.geeksforgeeks.org/python/morse-code-translator-python>

- **Eigene GitHub-Implementierung:**
https://github.com/Arshia-R13/Morse_Code