

Real-Time Emotion Recognition Across Cultures Using DeepFace and Haar Cascades

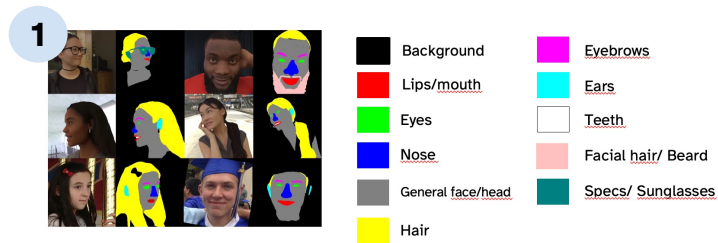
S. Cunha, Estefania

Bild 2 - WiSe 2025

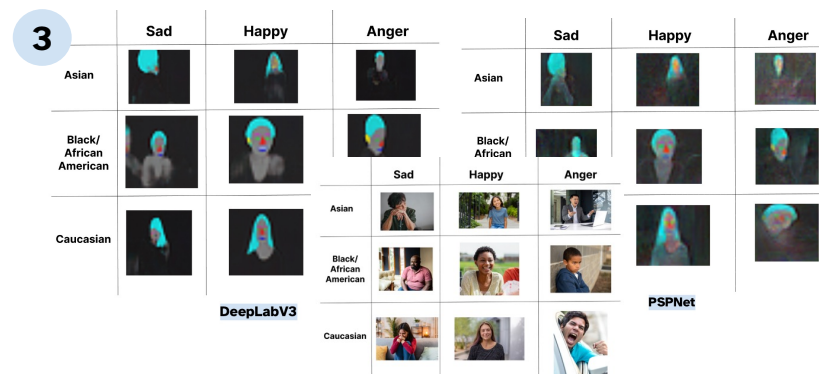
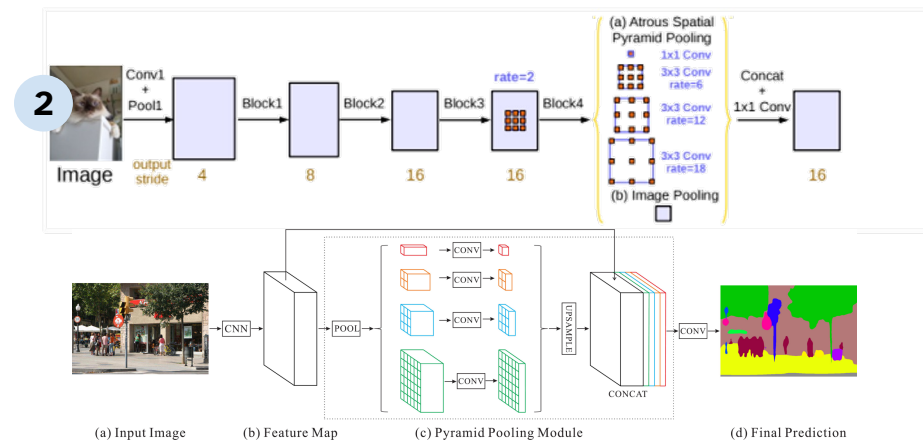
Agenda

1. Rückblick auf das vorherige Semester (Bild 1)
2. Real-Time Emotion Recognition (Bild 2)
3. Haar Cascades
4. DeepFace
5. Implementierung des Codes
6. Live Demonstration
7. Fazit & Diskussion

Rückblick – Bild 1



- MutIny-Datensatz mit **16.557** vollständig mit Pixeln beschrifteten Segmentierungsbildern.
- Enthält verschiedene **Ethnien, Altersgruppen** und **Geschlechter**.
- Ausgewogener Datensatz mit verschiedenen **Gesichtsposen** und **Kamerawinkeln**.



Real-Time Emotion Recognition

Ziel

- Emotionserkennung bei Menschen verschiedener Kulturen

Relevanz

- KI-Assistenten, Mental Health, Sicherheit

Methode

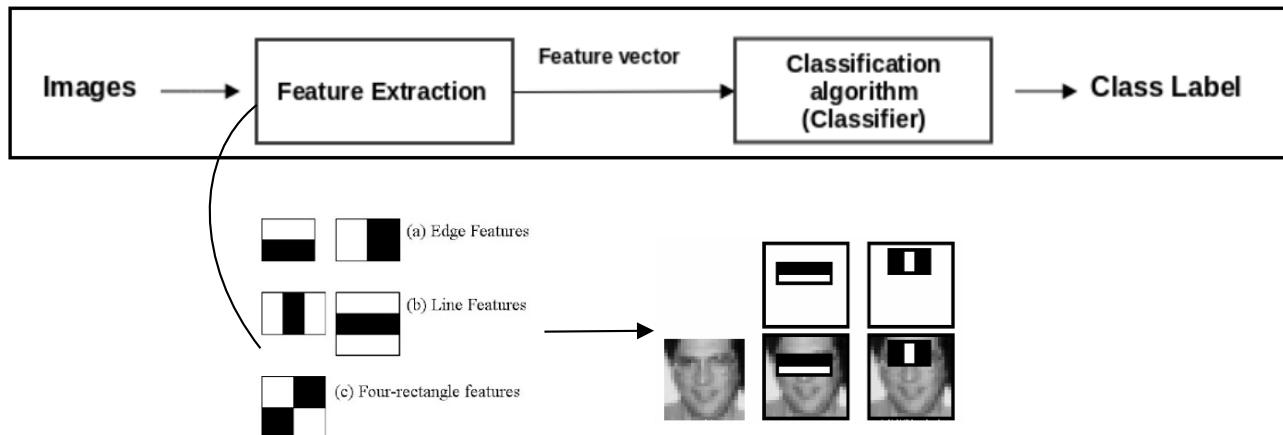
- Deep Learning (DeepFace) & Feature-Based CV (Haar-Cascades) → [GitHub-Projekt](#)

Haar Cascades

Entwicklung

- Erfunden 2001 von Paul Viola & Michael Jones
- Methode zur schnellen Objekterkennung in Bildern

Das Prinzip



OpenCV hat bereits fertige Haar-Cascades im System gespeichert. Diese kann man direkt benutzen.

DeepFace

DeepFace

- Eine Open-Source-Bibliothek für Gesichtserkennung & Gesichtsanalyse mit Deep Learning (CNNs)
- Entwickelt von Facebook AI (2014), trainiert mit 4 Mio. Gesichtern (nicht öffentlich)
- 97,35% Genauigkeit (nahe menschlicher Leistung)

Hauptfunktionen

- Face Verification, Face Recognition, Face Attribute Analysis, Echtzeit-Analyse

Implementierung des Codes I.

```
import cv2
from deepface import DeepFace

# Load face cascade classifier
#functions as a face finder, uses pre trained rules to detect faces in the webcam feed
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Start capturing video
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        print("Failed to capture frame. Exiting...")
        break

    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

    for (x, y, w, h) in faces:
        face_roi = frame[y:y + h, x:x + w]

        try:
            analysis = DeepFace.analyze(face_roi, actions=['emotion'], enforce_detection=False)

            # Extract key data
            emotions = analysis[0]['emotion']
            dominant_emotion = analysis[0]['dominant_emotion']
```

Implementierung des Codes II.

```
#
face_confidence = analysis[0].get('face_confidence', None)

# Display results
print(f"Detected emotions: {emotions}")
print(f"Dominant emotion: {dominant_emotion}")
print(f"Face confidence: {face_confidence:.2f}" if face_confidence else "Face confidence not available")

# Annotate frame with results
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
cv2.putText(frame, text: f"{dominant_emotion} ({emotions[dominant_emotion]:.2f}%)",
            org: (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, fontScale: 0.8, color: (0, 0, 255), thickness: 2)

except Exception as e:
    print(f"Error analyzing face: {e}")

# Display the resulting frame
cv2.imshow(winname: 'Real-time Emotion Detection', frame)

# Press 'q' to exit
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
```


Live Demonstration



Diskussion

Semantic Segmentation (Bild 1):

Pro

- Kein a priori Wissen über Objekteigenschaften erforderlich
- Keine Auswahl von Merkmalen erforderlich

Con

- Keine Transparenz → Fehlzuordnung
- Hoher Bedarf an Daten & Rechenleistung
- Lange Trainingszeiten, schwer nachvollziehbare Fehler

Feature-Based Computer Vision (Bild 2):

Pro

- Keine Trainingsdaten nötig, schnelle Implementierung
- Weniger Rechenleistung erforderlich.
- Erklärbarkeit und Interpretierbarkeit

Con

- Probleme mit Brillen & Lichtverhältnissen
- Weniger flexibel als Deep Learning

Quellenverzeichnis

1. Serengil, S. (n.d.). *DeepFace* [GitHub repository]. GitHub. Retrieved January 19, 2025, from <https://github.com/serengil/deepface>
2. Viso. (n.d.). *DeepFace: The Facebook deep learning facial recognition system*. Viso.ai. Retrieved January 19, 2025, from <https://viso.ai/computer-vision/deepface>
3. FH Bielefeld. (n.d.). *Gesichtserkennung: Haar-Cascades*. GitBooks. Retrieved January 19, 2025, from <https://fh-bielefeld-mif-sw-engineerin.gitbooks.io/script/content/embedded-computing/gesichtserkennung/haar-cascades.html>
4. Ajitharunai. (n.d.). *Facial emotion recognition with OpenCV and DeepFace* [Python script]. GitHub. Retrieved January 19, 2025, from <https://github.com/ajitharunai/Facial-Emotion-Recognition-with-OpenCV-and-Deepface/blob/main/emotion.py>