

# Spiegel AI facial recognition & Schnittstelle

MARCO KUNER – 03.07.2024

DATENVERARBEITUNG IN DER TECHNIK

# Projektziel

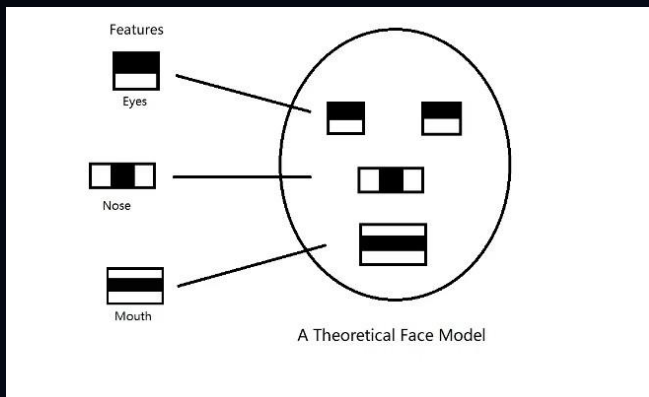
- Implementierung einer robusten Gesichtserkennung
- Erstellen von Profilen
- Wiedererkennung von Personen

# Grundlagen der Gesichtserkennung

- Face Detection
- Feature Extraction
- Matching and Recognition

# HAAR-Cascades

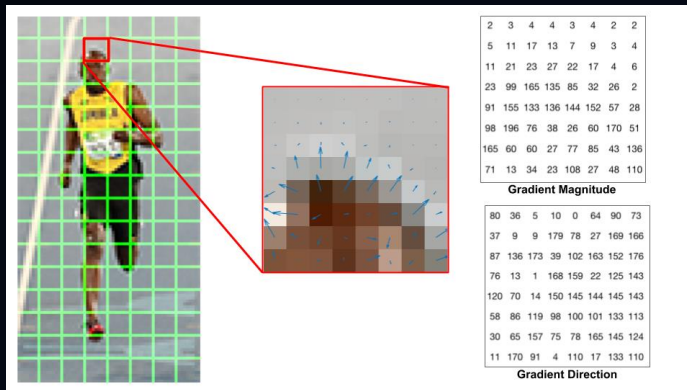
- Vorteil: Schnell und einfache Implementierung
- Nachteil: Begrenzte Genauigkeit



<https://st.quantrimang.com/photos/image/2022/11/11/lap-trinh-cong-cu-phat-hien-khuon-mat-bang-python-1.jpg>

# dlib

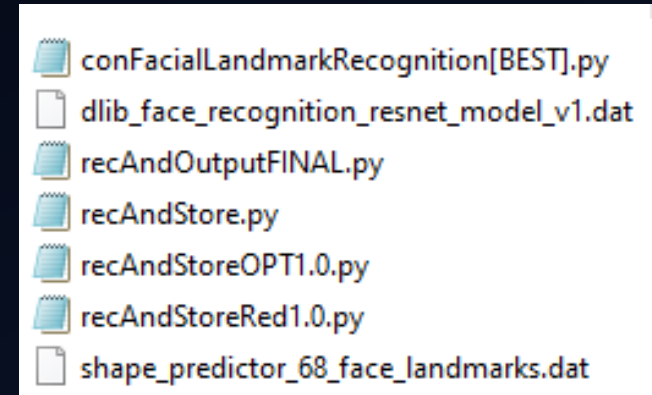
- Vorteil: Sehr präzise
- Nachteil: Umfangreich



<https://th.bing.com/th/id/R.a09546cec1edf3a919e28c64b9995970?rik=CFUFyTZKdo5%2fMQ&riu=http%3a%2f%2fjywiki.cn%2fpages%2fOS%2fimg%2fhog-cell-gradients.png&ehk=ywpOQaU1EEzX1DRuQasxIva0ABx105msq0JsXrDHI%3d&risl=&pid=imgRaw&r=0>

# Performance Troubleshooting

- Ziel: Verbesserung der dlib Performance
- Lösungsansätze:
  - Reduzierung der Bildgröße
  - Frame-Skips
  - Multithreading
  - Hardware-Beschleunigung
- Lösung?



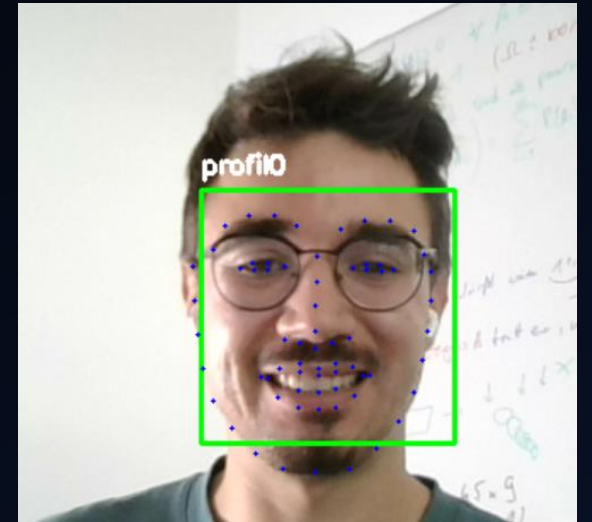
# Feature Extraction & Matching

- Gesichtspunkte extrahieren
- Face Embedding
- Matching erkannter Gesichter mit Database

```
# Load models
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
facerec = dlib.face_recognition_model_v1('dlib_face_recognition_resnet_model_v1.dat')
```

# Profile

- Werden sauber erkannt und geladen, aber:
- Ständige Synchronisation notwendig  
-> Nutzung des Websockets



```
def load_face_data():  
    global face_descriptors, labels, label_index  
    if os.path.exists('face_data.npz'):  
        data = np.load('face_data.npz', allow_pickle=True)  
        face_descriptors = data['face_descriptors'].tolist()  
        labels = data['labels'].tolist()  
        label_index = len(labels)
```

```
new_profile = {  
    "id": str(int(time.time() * 1000)),  
    "name": label,  
    "isSelected": True,  
    "selectedWidgetIDs": list(range(8)),  
    "selectedRemoteContent": [{"index": i, "id": i, "enabled": True} for i in range(9)]  
}  
profiles.append(new_profile)
```