



Westfälische
Wilhelms-Universität
Münster

Emotionserkennung

Klassifikation von Action-Units anhand von Landmarks



Inhalt

Aufgabenstellung

Programmierungumgebung

Pipeline

Evaluation



Inhalt

Aufgabenstellung

Eingabedaten

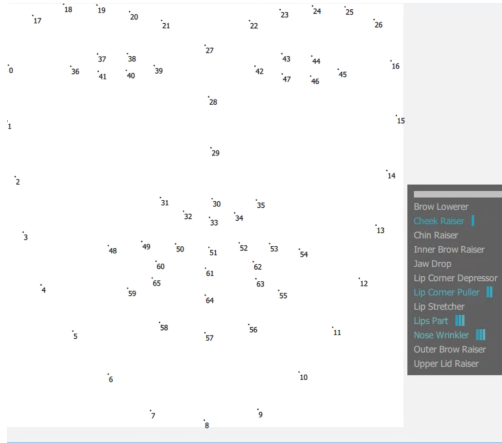
Ziel

Programmierungsumgebung

Pipeline

Evaluation

Eingabedaten



Ziel

Ziel

Trainieren eines Klassifikators, der in der Lage ist aus eingehenden Landmarks die aktivierten Action Units zu erkennen.



Inhalt

Aufgabenstellung

Programmierungumgebung

Pipeline

Evaluation



Programmierungsumgebung

- ▶ C++11
- ▶ OpenCV2
- ▶ QT5

Inhalt

Aufgabenstellung

Programmierungsumgebung

Pipeline

- Normalisierung

- Feature Extraction

- PCA

- Feature-Manipulation and -Scaling

- Klassifikation

- Zusammenfassung

Evaluation

Normalisierung

- ▶ Problem: Variationen von Position, Skalierung und Rotation in den Eingabevideos
- ▶ Lösung: Normalisierung der Daten
 - ▶ Zentrieren des Mittelpunktes
 - ▶ Skalierung zwischen 0..1
 - ▶ Rotation mithilfe der Augen

Feature Extraction

- ▶ Entwicklung verschiedener Merkmale aus den Landmark-Daten
- ▶ Beispiel: Relation der Landmarks untereinander
- ▶ Aufgabe: Extrahierung möglichst aussagekräftiger Merkmale
- ▶ Schwierigkeit: Aussagekraft der Merkmale vor dem Training unbekannt

Feature Extraction

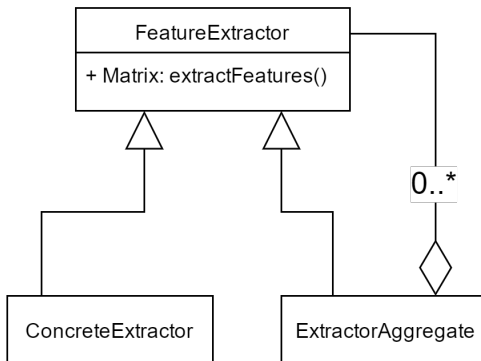


Abbildung: FeatureExtractor als Aggregate

Feature Extraction

- ▶ Unsere Features:
 - ▶ X-/Y-Koordinaten
 - ▶ Orientierung
 - ▶ Paarweise
 - ▶ Zum Mittelpunkt
 - ▶ Euklidische Distanz
 - ▶ Paarweise
 - ▶ Zum Mittelpunkt
 - ▶ Interpolation
 - ▶ Zeitbasiert

PCA

- ▶ $(66 \text{ Landmarks}) \times (\# \text{Features}) = \text{teilweise hohe Komplexität}$
→ Reduktion der Features auf Hauptkomponenten
- ▶ Approximation der Daten bei Erhaltung von 97% der Varianz
- ▶ Vorsicht: PCA betrachtet die Aussagekraft der Hauptkomponenten bezogen auf die Landmarks, NICHT auf die Erkennung der Action-Units

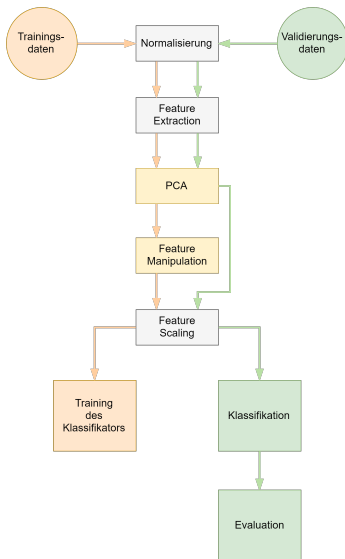
Feature-Manipulation and -Scaling

- ▶ Wenig true-positives in den Eingabedaten
→Optional: Generierung zusätzlicher true-positives
- ▶ Features mit unterschiedlichen Wertebereichen: Rotationsgrad, Distance,...
→Normalisierung der Features auf 0..1

Klassifikation

- ▶ Bearbeitete Features werden zum Training an Klassifikatoren übergeben
- ▶ Klassifikatoren können mit verschiedenen Parametern trainiert werden
- ▶ Problem erneut: optimale Parameter unbekannt
- ▶ Unsere Klassifikatoren:
 - ▶ Support Vector Machine: OpenCV::SVM
 - ▶ Random Forest: OpenCV::CvRTrees

Zusammenfassung



Inhalt

Aufgabenstellung

Programmierungumgebung

Pipeline

Evaluation

Methodik

Ergebnisse

Ausblick

Methodik

- ▶ Erstelle eine Konfigurationsdatei, in der alle zu testenden Klassifikatoren und Parameter gespeichert sind (Demo)
- ▶ Teile ersten Datensatz in 60% Trainings- und 40% Validierungsdaten auf
- ▶ Trainiere und evaluiere automatisch alle Klassifikatoren auf allen Action Units
- ▶ Wähle pro Action Unit die besten 5 anhand des F1 score aus
- ▶ Evaluiere Performance auf zweitem Datensatz (bisher unbekannte Testdaten)



Ergebnisse

Webdemo

Ergebnisse

| Klassifikator | Precision | Recall |
|---------------|-----------|--------|
| SVM | 0.6 | 0.5 |
| Random Forest | 0.8 | 0.7 |

Tabelle: Vergleich der Klassifikatoren

Ergebnisse

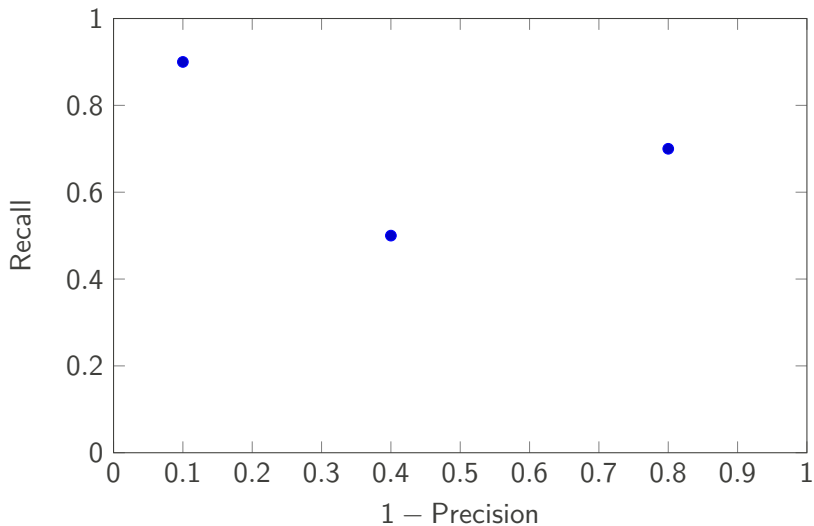


Abbildung: Precision-Recall-Kurve

Ausblick

- ▶ Verbesserung der Klassifikation durch Kombination von Klassifikatoren
- ▶ Ausbau der Time-Based Extraction
- ▶ ...