



Westfälische
Wilhelms-Universität
Münster

Emotionserkennung

Klassifikation von Action-Units anhand von Landmarks

Inhalt

Aufgabenstellung

Eingabedaten

Ziel

Programmierungumgebung

Pipeline

Normalisierung

Feature Extraction

PCA

Feature-Scaling and -Manipulation

Klassifikation

Zusammenfassung

Evaluation

Methodik

Ergebnisse

Vorausblick

Inhalt

Aufgabenstellung

Eingabedaten

Ziel

Programmierungsumgebung

Pipeline

Normalisierung

Feature Extraction

PCA

Feature-Scaling and -Manipulation

Klassifikation

Zusammenfassung

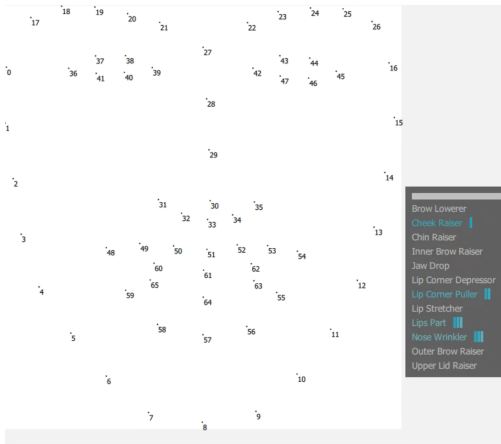
Evaluation

Methodik

Ergebnisse

Vorausblick

Eingabedaten



Ziel

Ziel

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

Inhalt

Aufgabenstellung

Eingabedaten

Ziel

Programmierungumgebung

Pipeline

Normalisierung

Feature Extraction

PCA

Feature-Scaling and -Manipulation

Klassifikation

Zusammenfassung

Evaluation

Methodik

Ergebnisse

Vorausblick



Programmierungsumgebung

- ▶ C++11
- ▶ OpenCV
- ▶ QT5

Inhalt

Aufgabenstellung

Eingabedaten

Ziel

Programmierungsumgebung

Pipeline

Normalisierung

Feature Extraction

PCA

Feature-Scaling and -Manipulation

Klassifikation

Zusammenfassung

Evaluation

Methodik

Ergebnisse

Vorausblick

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

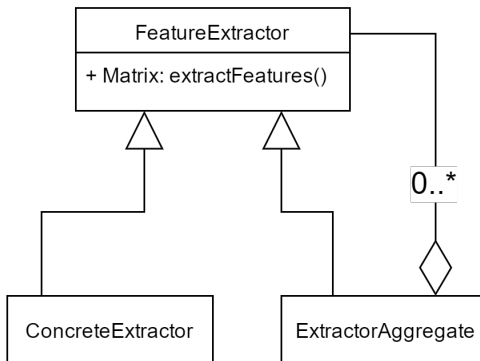


Abbildung: FeatureExtractor als Aggregate

Feature Extraction

- ▶ Unsere Features:
 - ▶ XYFeatureExtraction
 - ▶ OrientationExtraction
 - ▶ DistanceExtraction
 - ▶ TimeFeatureExtraction

PCA

- ▶ $(66 \text{ Landmarks}) \times (\# \text{Features}) = \text{teilweise hohe Komplexität}$
→ Reduktion der Frames auf besonders relevante Features

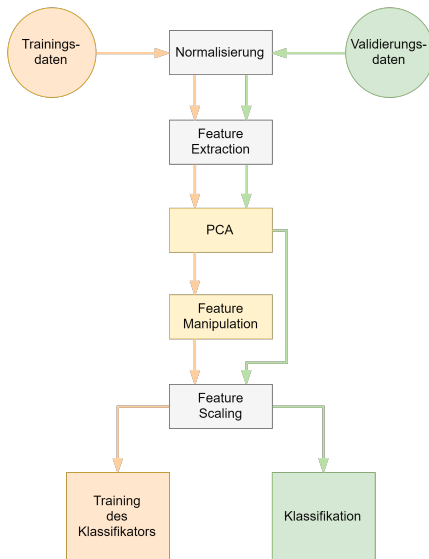
Feature-Scaling and -Manipulation

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

Klassifikation

- ▶ Klassifikatoren können mit verschiedenen Parametern trainiert werden
- ▶ Problem erneut: optimale Parameter unbekannt
- ▶ Unsere Klassifikatoren:
 - ▶ Support Vector Machine: `OpenCV::SVM` mit verschiedenen Parametern
 - ▶ Random Forest: `OpenCV::CvRTrees`

Zusammenfassung



Inhalt

Aufgabenstellung

Eingabedaten

Ziel

Programmierungumgebung

Pipeline

Normalisierung

Feature Extraction

PCA

Feature-Scaling and -Manipulation

Klassifikation

Zusammenfassung

Evaluation

Methodik

Ergebnisse

Vorausblick

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

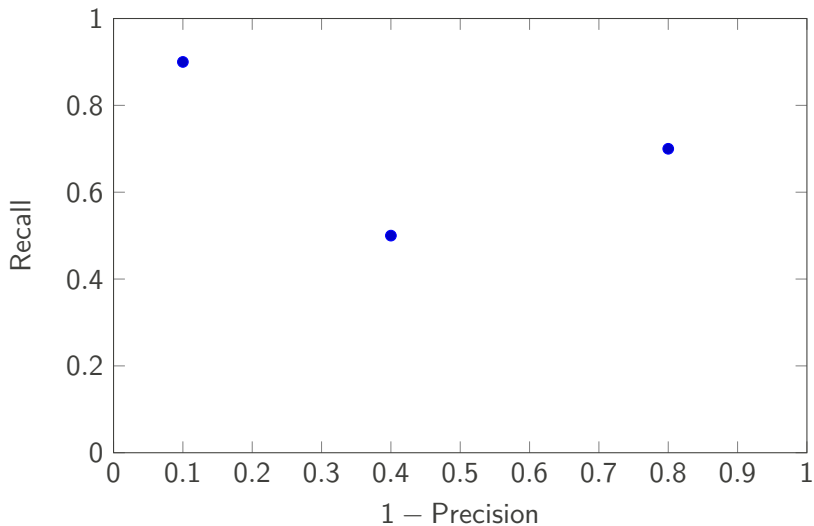


Abbildung: Precision-Recall-Kurve

Vorausblick

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