



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Emotionserkennung

Klassifikation von Action-Units anhand von Landmarks

Inhalt

Aufgabenstellung

Programmierungumgebung

Pipeline

Evaluation

Inhalt

Aufgabenstellung

Eingabedaten

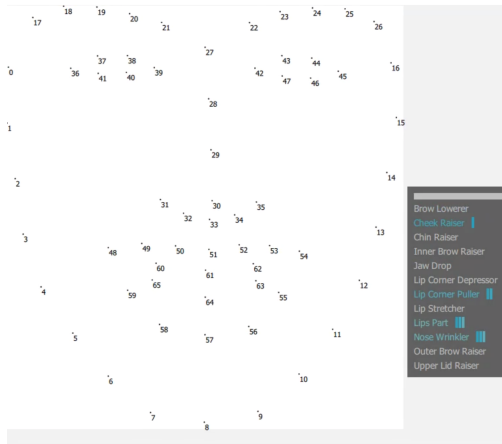
Ziel

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-Scaling and -Manipulation

- 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

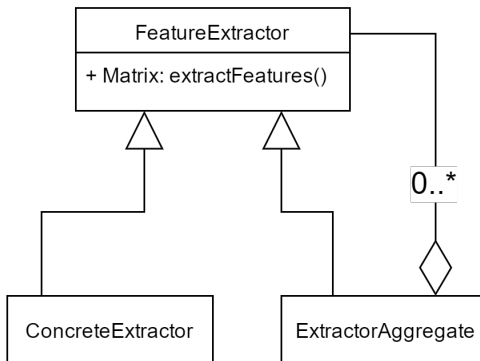


Abbildung: FeatureExtractor als Aggregate

Feature Extraction

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

PCA

- ▶ $(66 \text{ Landmarks}) \times (\# \text{Features}) = \text{teilweise hohe Komplexität}$
→ Reduktion der Frames auf besonders relevante Features
- ▶ ActionUnits betrachten isolierten Teil des Gesichts
→ Verwerfen irrelevanter Daten, die Störungen verursachen können

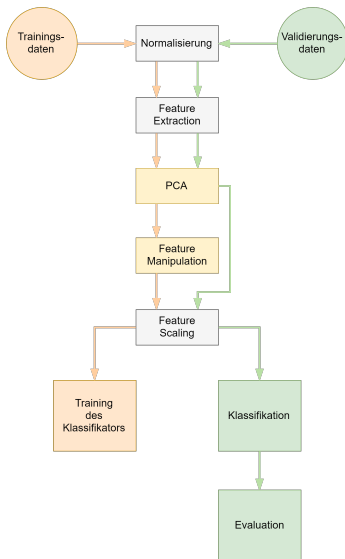
Feature-Scaling and -Manipulation

- ▶ 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`
 - ▶ Random Forest: `OpenCV::CvRTrees`

Zusammenfassung



Inhalt

Aufgabenstellung

Programmierungumgebung

Pipeline

Evaluation

Methodik

Ergebnisse

Diskussion

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

Diskussion

- ▶ Gute Ergebnisse bei “Lips Part”, akzeptable bei “Lip Corner Puller”
 - ▶ Möglicher Grund: vergleichsweise viele positive Daten (“Lips Part”: 8%)
 - ▶ Einfache Klassifikation anhand der Abstände der Mund-Landmarks
- ▶ Schlecht z.B. “Lip Corner Depressor”: $< 0.1\%$ positiv

Diskussion

- ▶ Trade-off zwischen Precision und Recall deutlich zu sehen
- ▶ Klassifikation mit Zeitableitung liefert keine guten Ergebnisse
 - ▶ Noch weniger positive Daten
- ▶ Random Forests erfolgreich zusammen mit CenterDistanceExtraction, InterpolationFeatureExtraction

Ausblick

- ▶ Verbesserung der Klassifikation durch Kombination von Klassifikatoren
- ▶ Erkennung von Emotionen, die sich aus mehreren Action Units zusammensetzen
- ▶ Ausbau der zeitbasierten Features
- ▶ Weitere Methoden zur Generierung positiver Samples



Ende