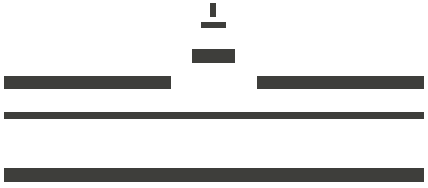


Westfälische
Wilhelms-Universität
Münster

Emotionserkennung

Klassifikation von Action Units anhand von Landmarks

wissen.leben
WWUMünster

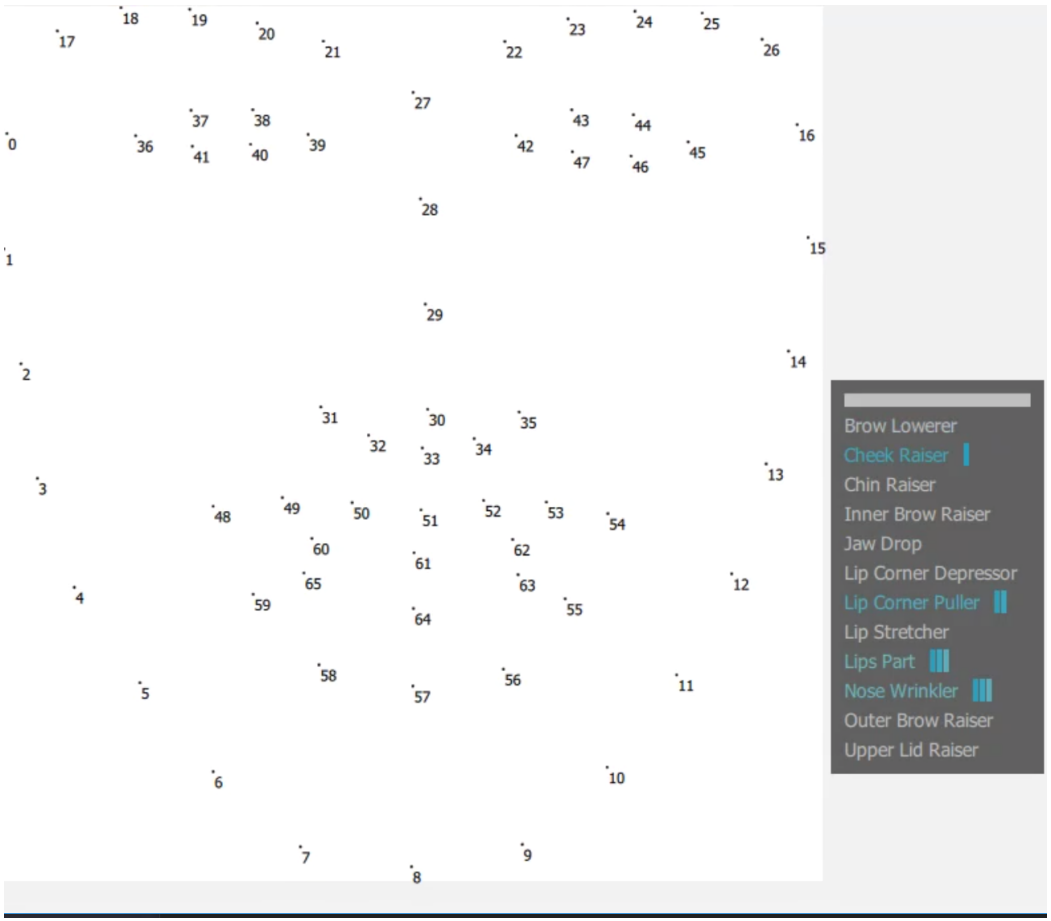
Robin, Johannes, Alexander

8. September 2016

Inhalt

Inhalt

Eingabedaten



Ziel

Ziel

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

Inhalt

Programmierumgebung

- C++11
- OpenCV2
- QT5

Inhalt

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

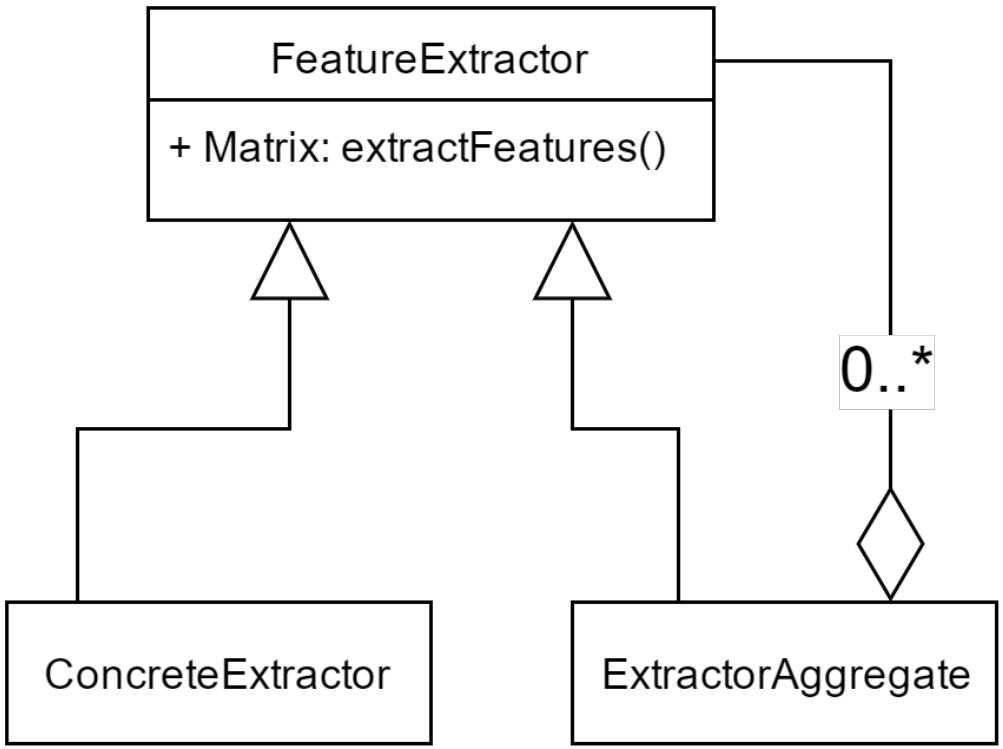


Abbildung: FeatureExtractor als Aggregate

Feature Extraction

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

PCA

- (66 Landmarks) $\times$ (#Features) = teilweise hohe Komplexität
  $\rightarrow$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

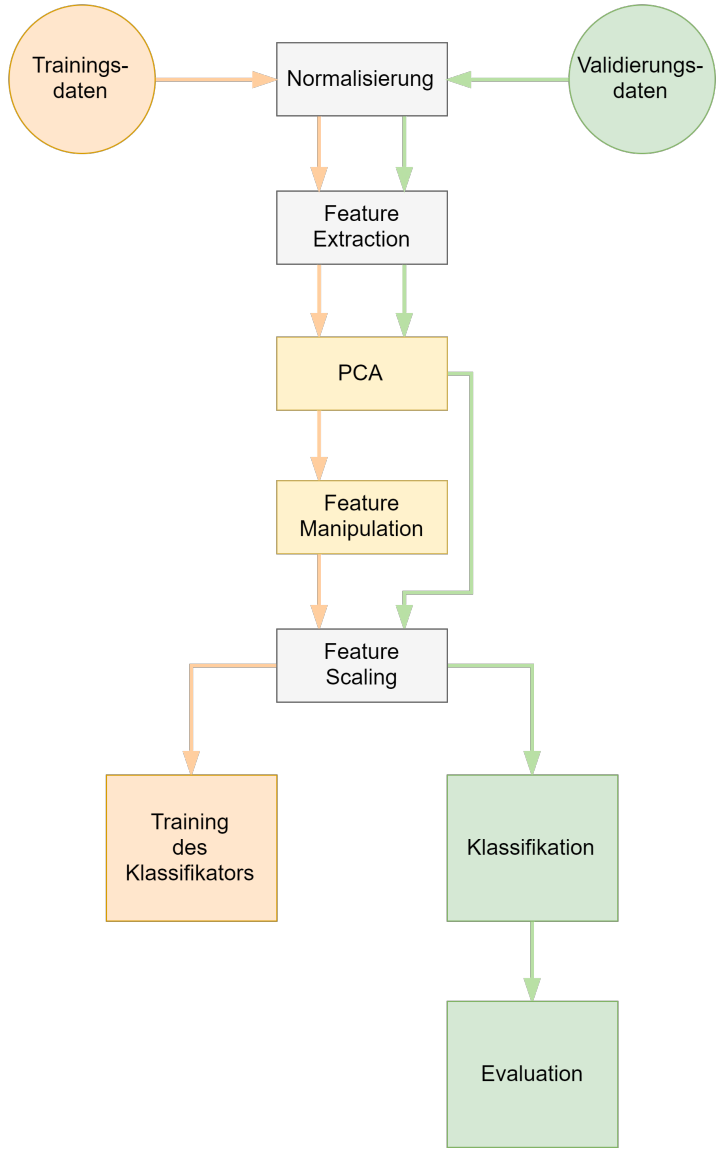
Feature-Manipulation and -Scaling

- 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

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