Westfälische Wilhelms-Universität Münster

Praktikum zur Mustererkennung

Emotionserkennung

Klassifikation von Action Units anhand von Landmarks

Robin Rexeisen
Matrikelnr. 123456

Johannes Stricker
johannesstricker@gmx.net
Matrikelnr. 383779

Alexander Schlüter
alx.schlueter@gmail.com
Matrikelnr. 409649

Betreuer:
Sören Klemm
soeren.klemm@wwu.de

eingereicht am 16. September 2016

Fachbereich 10
Mathematik und
Informatik

Vorwort

Hier entsteht ein Vorwort.

Inhaltsverzeichnis

1 Einleitung

TODO

1. Einführung in das Problems
2. Was für Daten wir zur verfügung (float-array) haben, woher die kommen (link auf Paper o.ä.).
3. Aufbau der Datenbank (mehrere Personen, ein Länge des Videos, ...)
4. Action-Units beschreiben (nur die in den Daten vorkommen)

2 Methodik

Hier keine Details zur Implementierung!

2.1 Vorverarbeitung

2.1.1 Punktwolken

1. Beschreibung der Daten ?!
2. Normalisierung
3. Randomisiertes erweitern
4. PCA

2.1.2 Feature Extrahieren

Statische Features

Dazu schreiben, wieso wir finden, dass das Feature ein gutes ist (z.B. weil es den Feature-Raum verkleinert, ...)

1. XYFeature
2. Orientation
3. EuclidianDistance
4. CenterDistance
5. CenterOrienation
6. Interpolation

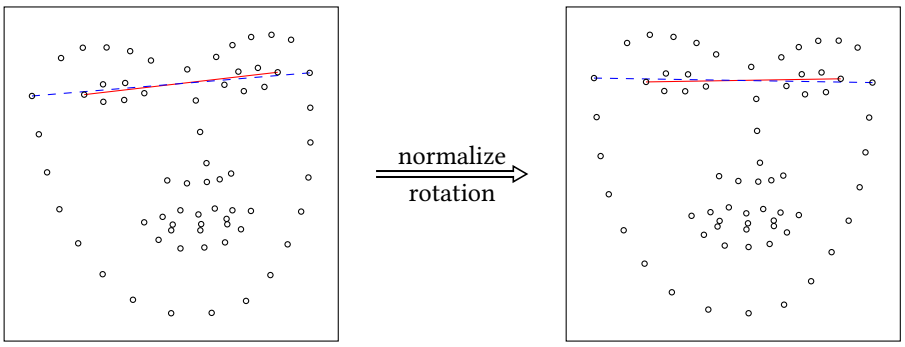

Abbildung 2.1: Normalisierung der Rotation anhand ausgesuchter Linien

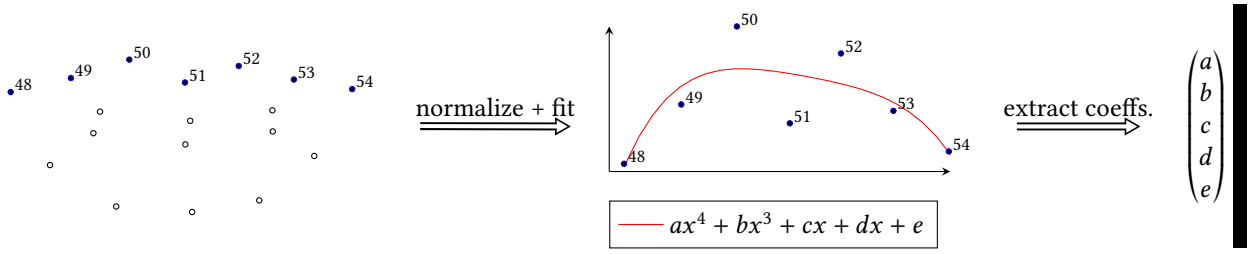

Abbildung 2.2: InterpolationFeatureExtraction

Zeitliche Features

- TimeDifferential

Featureverarbeitung

Zweck mitbeschreiben (z.B. PCA -> FeatureRaum weiter reduzieren)

- Negativanteil verringern
- MinMax/MeanVar Normalisieren
- Shufflen
- PCA

Klassifikatoren

SVM + Random Forests, Art von Parametern
Vielleicht noch eine Beschreibung einer allgemeinen Pipeline.

2.2 Evaluierungsmethoden

Die im vorherigen Abschnitt beschrieben Methoden zur Feature Extraction, Verarbeitung und Klassifikation sollen in verschiedenen Kombinationen evaluiert werden. Der erste Datensatz aus 10 Personen wird dazu aufgeteilt in 60% Trainingsmenge und 40% Validierungsmenge. Hier ist die Entscheidung zu treffen, wie die Personen auf die Mengen aufgeteilt werden:

1. Erst die Frames durchmischen, dann aufteilen: Dies ist sinnvoll, wenn der Klassifikator nur verwendet werden soll, um Action Units in neuen Frames von schon bekannten Personen zu erkennen. Es wird nicht getestet, wie gut der Klassifikator auf neue Personen generalisiert!

2. 6 Personen nur im Training, 4 nur in der Validierung verwenden: Die Performance auf der Validierungsmenge ist repräsentativ dafür, wie gut der Klassifikator Action Units bei bisher unbekannten Personen erkennt

Erste Tests haben gezeigt, dass Methode 1 zu deutlich besserern Performancestatistiken führt. Wir haben uns aber für Methode 2 entschieden, weil die Generalisierung auf neue Personen das interessantere Problem ist: In Anwendungsfällen ist es wünschenswert, für neue Personen nicht erst mehrere tausend Frames manuell labeln zu müssen, um den Klassifikator auf dieser Person zu trainieren.

Aufgrund der geringen Anzahl positiver Samples (Frames, in denen die Action Unit aktiviert ist), ist die Accuracy keine zuverlässige Statistik. Ein Klassifikator, der die Action Unit immer als “nicht aktiv”

klassifiziert, könnte sehr hohe Accuracy erreicht, ohne tatsächlich etwas über die Action Unit gelernt zu haben. Stattdessen evaluieren wir die Klassifikatoren anhand von

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}, \qquad \text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Der F1 score ist das harmonische Mittel zwischen Precision und Recall. Da wir von der Aufgabenstellung her keine Präferenz für hohe Precision / hohen Recall haben, nutzen wir den F1 score als erste Zahl zum Vergleich der Klassifikatoren.

Es werden alle Kombinationen aus Feature Extraction, Klassifikator und Parametern auf dem ersten Datensatz trainiert und evaluiert. Anschließend werden die besten fünf pro Action Unit anhand des F1 scores ausgewählt. Da pro Action Unit ca. 160 Kombinationen evaluiert werden, kann es durch diese Auswahl der besten fünf zu einem Overfitting gegen die Validierungsmenge kommen. Um realistische Zahlen für die Performance zu bekommen, werden deshalb die besten fünf nochmal auf einer Testmenge evaluiert. Diese besteht aus fünf bisher unbekannten Personen aus einem zweiten Datensatz.

3 Implementierung

- Architektur/OS eines lauffähigen Systems
- Softwareabhängigkeiten
- Programmiersprache

3.1 QViewer

1. Zweck
2. Bilder

3.2 Auto-Train

- Zum trainieren und evaluieren
- Kurzes Wort zum Design von FeatureExtractor
- Automatisches Speichern aller relevanten Dateien.
- Erwähnung der JSON-Konfigurations-Datei
  - Design von Processors

4 Ergebnis

- Welche Parameter/Pipeline zum trainieren
  - Warum diese Parameter und keine anderen?
- Alle Plots zeigen (oder nur eine Teilmenge?)
  - Bei den Plots schwache Punkte gar nichts erst anzeigen?!
  - Auf jedenfall gute immer beschreiben (welche Parameter z.B.)
- Auflisten welche Action-Unit welche Classificator gut war (Recall, Precision, F1-Score)
- Klar machen, wie gut diese Klassifikatoren bei Trainingsmenge abschneiden
- Allgemeine Aussage, welche Klassifikatoren mit Parametern überhaupt nicht geeignet sind und welche super sind.
- Aussage welche Action-Unit gut zu klassifizieren ist
- Erwähnen, dass Shuffle SVM-Ergebnisse ändert.

AU	Bester Klassifikator				
	F1 Val	F1 Test	Precision Test	Recall Test	Features
Lip Corner Puller	0.37	0.431	0.317	0.674	XY
Outer Brow Raiser	0.391	–	0	0	EuclidianDistance
Lip Corner Depressor	0.022	–	0	0	XY
Upper Lid Raiser	0.394	–	0	0	Interpolation
Inner Brow Raiser	0.2	0.084	0.064	0.125	EuclidianDistance
Cheek Raiser	0.277	0.18	0.102	0.769	XY
Lips Part	0.656	0.575	0.598	0.554	CenterDistance
Brow Lowerer	0.163	0.2	0.659	0.118	EuclidianDistance
Chin Raiser	0.191	0.03	0.163	0.017	CenterDistance
Nose Wrinkler	0.242	0.03	0.015	0.62	EuclidianDistance

Abbildung 4.1: F1 scores und Testergebnisse des besten Klassifikators pro Action Unit

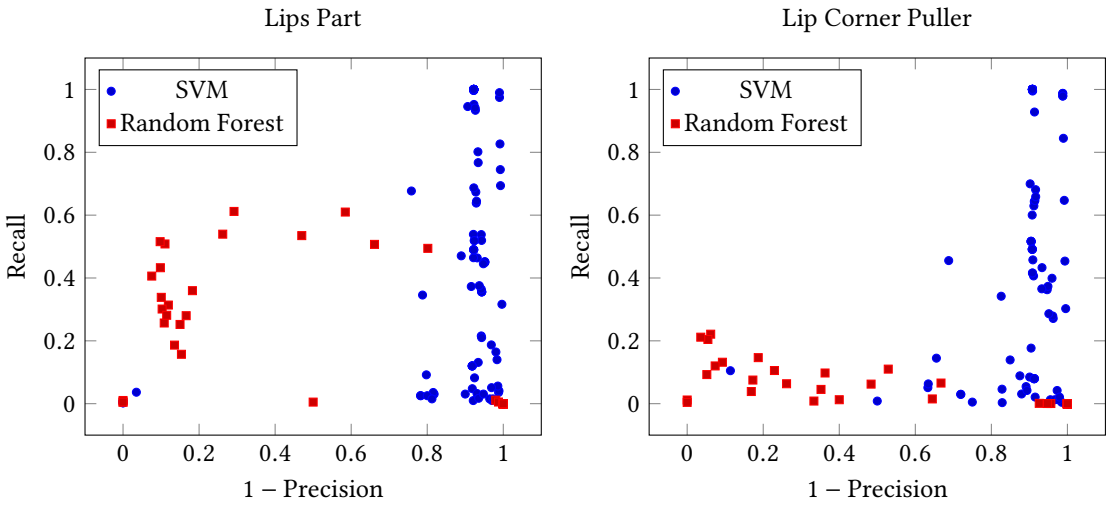

Abbildung 4.2: Ergebnisse der Klassifikatoren und Parameter für Lips Part

F1 score Validation	F1 score Test	Precision Test	Recall Test
0.656	0.575	0.598	0.554
0.656	0.594	0.639	0.555
0.647	0.655	0.809	0.55
0.623	0.616	0.966	0.452
0.585	0.732	0.752	0.713

Abbildung 4.3: F1 scores und Testergebnisse der Top 5 Klassifikatoren für Lips Part

5 Diskussion

- Wieso sind gut bei Test, aber schlecht bei Training
- Wieso sind diese Klassifikatoren gut und andere nicht
- Wieso sind viele Action-Units schlecht zu erkennen
- Wieso gerade diese Feature so gut?
- ⇒ Overfitting
- Warum geht diese Action-Unit besser als andere

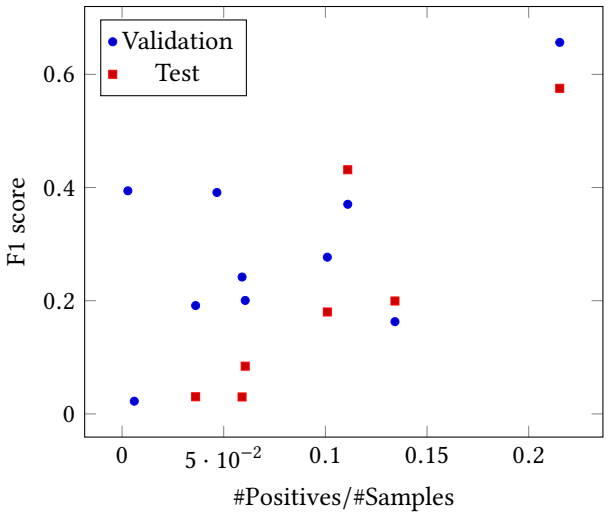

Abbildung 5.1: F1 score des besten Klassifikators jeder Action Unit gegen Anteil positiver Samples

6 Fazit

6.1 Zusammenfassung

6.2 Ausblick

- Was könnte man noch verbesser, und wieso haben wir das nicht gemacht (z.B. aus Zeitgründe)
    1. Mehr Kombinationen (Mit/ohne PCA, mehr Time-Differential-Feature, überhaupt mehr zeitliche Features, andere normalisierungen der Punktwolke, Neuronales-Netzwerk oder andere Klassifikatoren dazu benutzen)
- Wie könnte das Ergebnis besser werden (z.B. mehr Daten von mehreren Personen)