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

Das Thema der (visuellen) Emotionserkennung durch Computersysteme hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Die Einsatzgebiete sind vielseitig und reichen von Sicherheitsanwendungen, über Robotik, bis hin zu Unterhaltungsmedien. Meist wird versucht, anhand von verschiedenen Merkmalen im Gesicht, diesem eine oder mehrere Emotionen zuzuordnen. Im Rahmen unsere Praktikums, war es unsere Aufgabe ein solches Computersystem zur Erkennung von Emotionen zu entwickeln.
Im Folgenden Kapitel werden die Aufgabenstellung, sowie die Eingabedaten genauer beschrieben. In Kapitel 2 werden daraufhin die Methodiken vorgestellt, die wir für unser Programm nutzen, woraufhin in Kapitel 3 erläutert wird, wie wir diese implementiert haben. Daraufhin folgt die Vorstellung unserer Ergebnisse in Kapitel 4 und ein Ausblick auf mögliche Erweiterungen der Anwendung. Im letzten und 5. Kapitel wird das Ergebnis der Arbeit kurz resümiert.

1.1. Facial Action Code

Der Facial Action Code (kurz FAC) ist ein System zur Unterscheidung von Bewegungen von isolierten Teilen des menschlichen Gesichts, welches 1976 von Paul Ekman und Wallace V. Friesen entwickelt wurde. Es basiert auf sogenannten Action Units (kurz AU), welche eben genau diese Bewegungen beschreiben sollen. Dabei kann eine Action Unit eine Ausprägung zwischen einschließlich 0 und 5 haben, wobei 0 bedeutet, dass keine entsprechende Bewegung vorhanden ist, und 5 bedeutet, dass die Bewegung maximal stark ausgeprägt ist [EF76]. Eine Auflistung der für diese Arbeit relevanten Action Units findet sich im Anhang A.1.

1.2. DISFA Datenbank

Die Denver Intensity of Spontaneous Facial Action Database (kurz DISFA Database) enthält eine Sammlung von Gesichtsbewegungen von insgesamt 27 unterschiedlichen, erwachsenen Probanden. Hierzu wurde von jedem Probanden ein 4-minütiges Video mit je 20 Frames pro Sekunde gedreht. Danach wurde jedes Frame nach dem Facial Action Coding System auf die Ausprägung von 12 Action Units analysiert und gelabelled. Weiterhin enthält jedes Frame 66 Landmark Koordinaten, von markanten Punkten des Gesichtes.

1.3. Die Aufgabenstellung

Die Aufgabenstellung des Praktikums bestand darin, aus einer Auswahl von 12 Videos der DISFA Datenbank einen Klassifikator zu entwickeln, der möglichst präzise in der Lage ist für ein beliebiges Frame aus der Datenbank zu bestimmen, welche Action Units in dem Frame aktiviert sind (dh. eine Ausprägung größer oder gleich 1 haben).

2. Methodik

In diesem Kapitel werden sowohl die von uns verwendeten Methoden zum Training der Klassifikatoren und zum Klassifizieren, als auch die von uns verwendeten Klassifikatoren selbst genauer beschrieben. Außerdem wird erläutert, wie die Ergebnisse evaluiert wurden.

2.1. Vorverarbeitung

2.1.1. Aufbereitung der Eingabedaten

Wie bereits in der Einleitung erwähnt, handelt es sich bei den Eingabedaten um 12 Videos, die der DISFA Datenbank entnommen sind. Die Videos wurden nacheinander aufgenommen und zeigen verschiedene Probanden. Bedingt dadurch, sind die Landmarks in den Videos nicht identisch bezüglich Skalierung, Rotation und Position.
Damit die Klassifikation durch diese Störungen nicht beeinträchtigt wird, werden die Eingabedaten zunächst normalisiert. Dies geschieht in drei Schritten.

1. Die Landmarks werden um den Koordinaten-Ursprung zentriert. Hierzu berechnen wir einen Vektor vom Mittelpunkt aller Landmarks zum Ursprung und translatieren die gesamte Punktwolke um diesen Vektor.

2. Daraufhin wird die Punktwolke so skaliert, dass die maximale horizontale Distanz alle Landmarks genau 1 beträgt. Hierzu berechen wir diese maximale Distanz und teilen alle Koordinaten der Landmarks durch diese.

3. Um Störungen durch Drehung des Kopfes der Probanden auszugleichen, normalisieren wir ebenfalls die Rotation der Punktwolke. Dazu berechnen wir einen Vektor zwischen den beiden Augen des Probanden und rotieren die gesamte Punktwolke so, dass dieser Vektor auf eine Rotation von 0° gebracht wird 2.1.1

Ein weiteres Problem der Eingabedaten besteht darin, dass die Relation von true-positives zu true-negatives sehr gering ist, das heisst die Anzahl der Frames in denen eine bestimmte Action Unit aktiv ist, ist für die meisten Action Units relativ gesehen sehr gering.

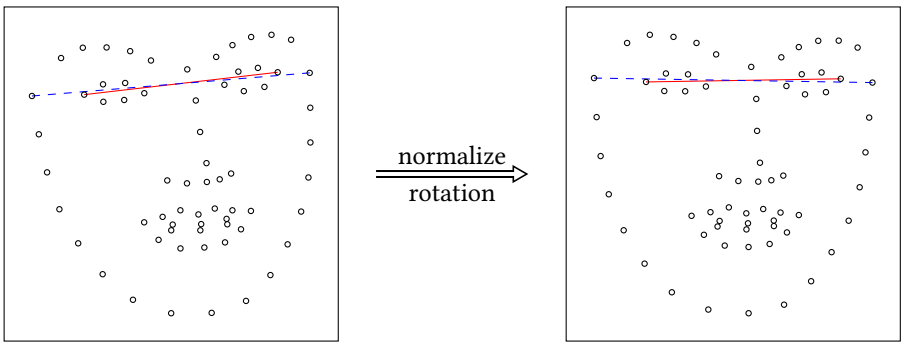

Abbildung 2.1.: Normalisierung der Rotation durch einen Vektor zwischen den Augen des Probanden

Um dieses Problem zu reduzieren, ermöglichen wir es die true-positives der Eingabedaten zu erweitern, indem die entsprechenden Frames dupliziert und die Landmarks in diesen Frames durch eine leichte, normalverteilte Störung verschoben werden.

2.1.2. Feature Extraction

Bei der (visuellen) Emotionserkennung wird versucht anhand von einem oder mehreren, verschiedenen Merkmalen (engl. features) einem Gesicht eine oder mehrere Emotionen zuzuordnen. Je mehr Aussagekraft die Kombination dieser Merkmale über die jeweiligen Emotionen haben, desto besser können diese klassifiziert werden. Das Problem dabei ist, dass meist weder die Merkmale, noch ihre Aussagekraft zuvor bekannt sind. Deshalb extrahieren wir aus den Eingabedaten, also den Videos mit je 66 Landmarks pro Frame, verschiedene Merkmale, um sie in verschiedenen Kombinationen miteinander zu testen. Es folgt eine Beschreibung der von uns verwendeten Features.

Statische Features

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

1. **X-/Y-Koordinaten**: die Koordinaten der Landmarks werden als Merkmale genutzt. Da in der Menge der Koordinaten sowohl Informationen über die individuellen Punkte liegen, als auch Informationen über ihre Relation zueinander, ist es sinnvoll dieses Feature zu testen.

2. **Paarweise Orientierung**: es werden jeweils alle Paare von je zwei unterschiedlichen Landmarks betrachtet und die Rotation des Vektors zwischen den beiden Punkten als Merkmal genutzt. Weil sich bei verschiedenen Mimiken meist die Position markanter Punkte im Gesicht zueinander ändert, erscheint es sinnvoll Features zu nutzen, die die Landmarks untereinander explizit in Relation setzen.

3. **Paarweise Euklidische Distanz**: auch hier werden jeweils alle Paare unterschiedlicher Landmarks betrachtet und die euklidische Distanz zwischen den beiden Punkten als Merkmal genutzt. Dieses Feature erscheint ebenfalls sinnvoll, weil es Informationen über die Relation von Landmarks untereinander hat.

4. **Orientierung relativ zum Mittelpunkt der Landmarks**: bei diesem Feature wird die Orientierung jedes Landmarks relativ zum Mittelpunkt aller Landmarks betrachtet, das heisst es wird die Rotation des Vektors zwischen Mittelpunkt und Landmarks als Merkmal genutzt. Dieses Feature enthält Informationen darüber, wie die Position der Landmarks relativ zum gesamten Gesicht ist. Dies erscheint für viele Gesichtsausdrücke sinnvoll.

5. **Euklidische Distanz zum Mittelpunkt der Landmarks**: dieses Feature betrachtet die euklidische Distanz jedes Landmarks zum Mittelpunkt aller Landmarks. Dieses Feature sagt ebenfalls etwas über die Relation der einzelnen Landmarks zum gesamten Gesicht aus.

6. **Polynominterpolation**: es wird versucht zusammenhängende Landmarks, das heisst Punkte, welche zusammen einen Teil des Gesichtes ergeben, durch ein Polynom zu interpolieren und die Polynomkoeffizienten als Feature zu extrahieren. Die Action Units beziehen sich meist auf genau einen isolierten Bereich des Gesichtes. Daher erscheint es naheliegend, diese Bereiche durch eine Funktion zu approximieren und diese als Feature zu nutzen.

Zeitliche Features

- TimeDifferential

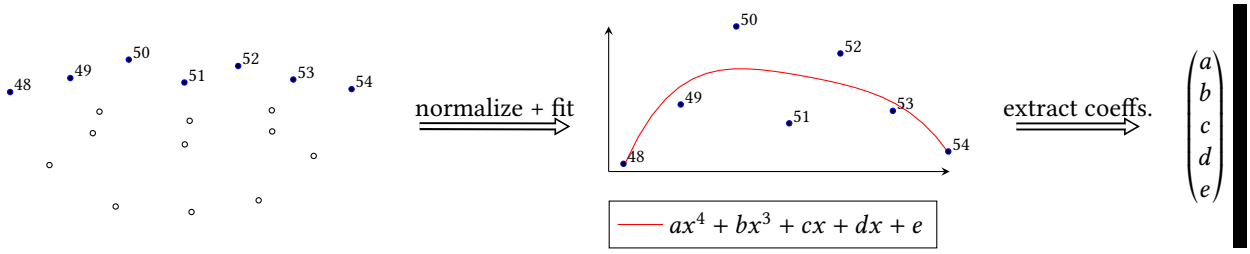

Abbildung 2.2.: InterpolationFeatureExtraction

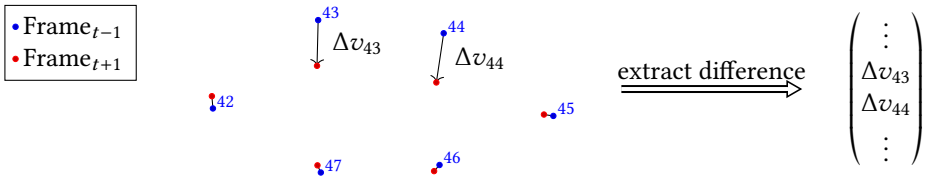

Abbildung 2.3.: TimeDifferentialExtraction

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.

- Welche Parameter/Pipeline zum trainieren
  - Warum diese Parameter und keine anderen?

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

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

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

In Tabelle 4.1 sind die Ergebnisse des besten Klassifikators (ausgewählt nach F1 score auf der Validierungsmenge) pro Action Unit zu sehen. Gute Klassifikation auf unbekannten Personen ist möglich für Lips Part: Der hohe F1 score 0.656 in der Validierung bestätigt sich auch auf der Testmenge. Akzeptable Performance liefert der beste Klassifikator für Lip Corner Puller. Dieser verbessert sich sogar von einem F1 score von 0.37 in der Validierung auf 0.431 im Test.

Die Klassifikatoren für Outer Brow Raiser und Upper Lid Raiser scheinen in der Validierung akzeptabel, erkennen jedoch im Test überhaupt keine Aktivierung der Action Units mehr. Die übrigen Action Units werden mit keiner unserer Feature Extraction-Methoden an neuen Personen befriedigend klassifiziert.

Abb. 4.1 zeigt, dass die verschiedenen Kombinationen aus Feature Extraction, Klassifikator und Parametern zu stark variierender Performance auf der Validierungsmenge führen. Der Tradeoff zwischen Precision und Recall ist deutlich zu sehen. Man sieht eine Trennung zwischen SVM und Random Forest Klassifikatoren: erstere tendieren dazu, zu viele negative Frames (ohne Aktivierung der Action Unit) als positiv zu klassifizieren, was zu schlechter Precision führt. Die Random Forests neigen hingegen zu vielen False Negatives. Sie schneiden bezogen auf Lips Part besser ab, bei anderen Action Units ist die Performance zwischen den Klassifikatoren ausgeglichen.

F1 Val	F1 Test	Prec. Test	Recall Test	Klassif.	Parameter	Features
0.656	0.575	0.598	0.554	RF	#trees = 20, maxDepth = 20	CenterDist.
0.656	0.594	0.639	0.555	RF	#trees = 50, maxDepth = 20	CenterDist.
0.647	0.655	0.809	0.550	RF	#trees = 20, maxDepth = 10	CenterDist.
0.623	0.616	0.966	0.452	RF	#trees = 50, maxDepth = 10	Interpolation
0.585	0.732	0.752	0.713	RF	#trees = 20, maxDepth = 20	XY

Tabelle 4.2.: F1 scores und Testergebnisse der Top 5 Klassifikatoren für Lips Part

Die Dominanz der Random Forests für Lips Part ist auch in Tabelle 4.3 zu sehen. Die beste SVM taucht mit einem F1 score von 0.356 in den Top 5 nicht mehr auf.

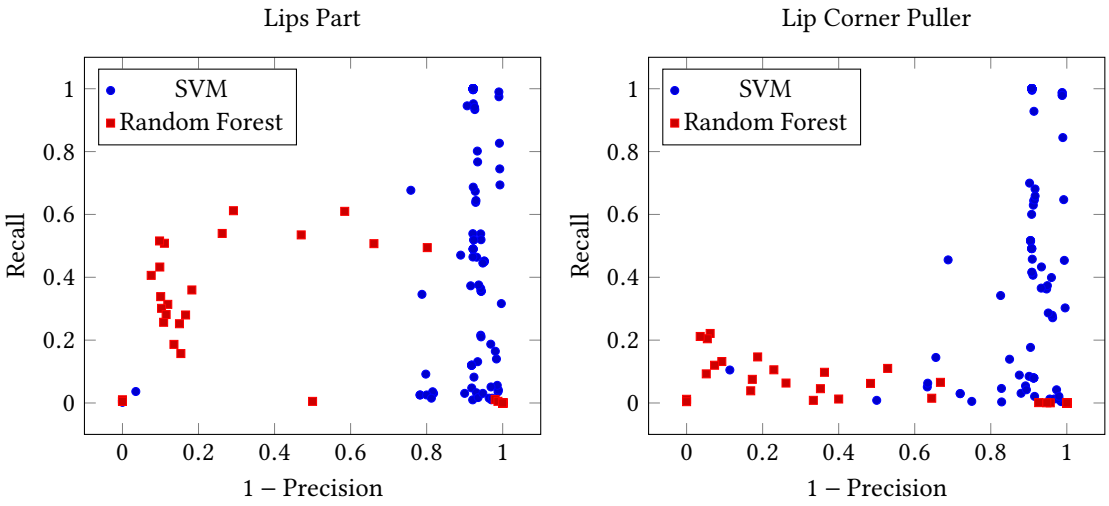

Abbildung 4.1.: Validierungsergebnisse für Lips Part und Lip Corner Puller. Jeder Punkt steht für eine Kombination aus Feature Extraction, Klassifikator und Parametern.

F1 Val	F1 Test	Prec. Test	Recall Test	Klassif.	Parameter	Features
0.370	0.431	0.317	0.674	SVM Lin.	–	XY
0.358	0.514	0.675	0.415	RF	#trees = 10, maxDepth = 4	Interpolation
0.347	0.389	0.752	0.262	RF	#trees = 50, maxDepth = 20	Interpolation
0.336	0.536	0.733	0.423	RF	#trees = 20, maxDepth = 10	Interpolation
0.248	0.426	0.774	0.293	RF	#trees = 20, maxDepth = 10	EuclidianDist.

Tabelle 4.3.: F1 scores und Testergebnisse der Top 5 Klassifikatoren für Lip Corner Puller

Features	Bester F1 score	Dimensionen	Dim. nach PCA
CenterDistance	0.656	66	17
Interpolation	0.623	37	–
XY	0.585	132	28
EuclidianDistance	0.5	2,145	30
Orientation	0.144	2,145	2,145
CenterOrientation	0.137	66	2
TimeDiff_XY	0.126	132	117
TimeDiff_Interpolation	0.06	37	–

Tabelle 4.4.: Vergleich der verschiedenen Features für Lips Part. Auf Interpolation-Features wurde keine PCA angewandt.

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?
- $\Rightarrow$ Overfitting
- Warum geht diese Action-Unit besser als andere

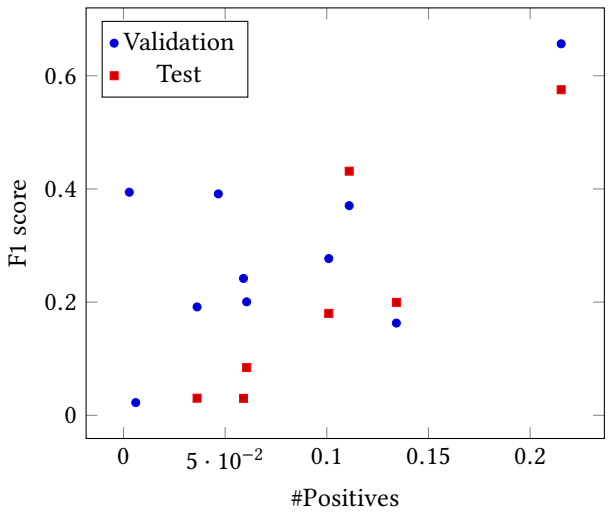

Abbildung 5.1.: F1 score des besten Klassifikators jeder Action Unit gegen Anteil positiver Samples

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)

A. Anhang

A.1. Auflistung der relevanten Action Units

1. Inner Brow Raiser
2. Outer Brow Raiser
3. Brow Lowerer
4. Upper Lid Raiser
5. Cheek Raiser
6. Lid Tightener
7. Nose Wrinkler
8. Upper Lid Raiser
9. Nasolabial Fold Deepener
10. Lip Corner Puller
11. Cheeck Puffer
12. Dimpler
13. Lip Corner Depressor
14. Lower Lip Depressor
15. Chin Raiser
16. Lip Puckerer
17. Lip Stretcher
18. Lip Funneler
19. Lip Tightner
20. Lip Pressor
21. Lips Part
22. Jaw Drop
23. Mouth Stretch
24. Lip Suc

Literatur

[EF76] Paul Ekman und Wallace V. Friesen. „Measuring Facial Movement“. In: *Environmental Psychology and Nonverbal Behavior* (1976) (siehe S. 1).