

Projektdokumentation Wissens- basierte Systeme

Thema: Evidenzen

Datensatz: 1

Autoren: Malte Bartels, Hendrik Ulbrich; TInf16C

Dozent: Prof. Dr. Dirk Reichardt

Abgabe: 10.04.2019

Inhaltsverzeichnis

1. Überblick	2
2. Verwendung	3
2.1 Voraussetzungen	3
2.2 Anwendung	3
2.3 Quellcode	4
3. Herangehensweise	5
3.1 Datenimport	5
3.2 Diskretisierung	5
3.3 Dempster-Shafer Regel	7
4. Detaildokumentation	9
4.1 CSV Model	9
4.2 Feature Model	10
4.3 FacialExpression Model	10
5. Verifikation	11

1. Überblick

Diese Anleitung beschreibt die Verwendung und Funktionsweise des Projekts *emotion-recognition*, welches im Rahmen der Vorlesung Wissensbasierte Systeme an der DHBW Stuttgart als Programmentwurf entwickelt wurde. Grundlegendes Ziel der Applikation ist es, anhand von .csv Dateien, welche pixelbasierte Messwerte verschiedener Gesichtspartien von Portraitfotos enthalten, Emotionen zu erkennen. Diese Anleitung stellt keine Dokumentation des Quellcodes dar. Eine solche Dokumentation befindet sich als JavaDoc unter *docs/index.html*. Zusätzlich finden sich an notwendigen Stellen Kommentare im Quellcode.

2. Verwendung

In diesem Abschnitt wird die produktive Verwendung der exportierten Applikation erklärt.

2.1 Voraussetzungen

Vorausgesetzt wird ein JDK 1.8_u141 oder höher. Die Anwendung wurde auf JDK 1.10_u2 getestet. Es handelt sich beim Buildartefakt um einen Export als .jar Datei mit folgendem Einsprungspunkt:

```
de.emotreco.main.EmotionRecognitionMain
```

Um eine Datei zu analysieren, muss diese dem Format ([RFC 4180](#)) der in der Aufgabe bereitgestellten Datensätze entsprechen. Das heißt auch, dass Spalten durch ; und Zeilen durch \n separiert werden müssen. Die Reihenfolge der Spalten muss konstant folgendem Schema der Kopfzeile folgen:

```
Nr;x;y;xright;ylow;fob;lea;lbd;rea;rbd;hnc;vnc;lcw;rcw;ma
```

Um zu indizieren, dass bei einem Frame für eine Spalte kein Wert vorliegt, muss die jeweilige Zelle keine Zeichen enthalten. Ferner enthält eine Datei immer 60 Frames, welche wie in der Aufgabenstellung (siehe *exercise/WBS_PE_Evidenzen.pdf*) angeordnet sind.

2.2 Anwendung

Das Starten der exportierten Applikation erfolgt über die Konsole durch den Befehl:

```
java -jar emotion-recognition.jar emo_muster_1_1.csv
```

Dabei wird die Datei emo_muster_1_1.csv analysiert. Der Dateiname kann durch einen beliebigen Dateipfad ersetzt werden. Ferner ist es möglich, mehrere Dateien hintereinander mit einem Befehl zu prüfen:

```
java -jar emotion-recognition.jar --all
```

Hier werden alle .csv Dateien aus dem Order data analysiert, welcher sich im selben Verzeichnis, wie die .jar Applikation befinden muss. Wird der Synopsis der Applikation nicht entsprochen, gibt eine Fehlermeldung die valide Syntax aus.

2.3 Quellcode

Der im Rahmen der Projektaufgabe erstellte Quellcode wurde in IntelliJ IDEA Version 2019.1 verfasst. Zusammen mit dem Quellcode wird der Ordner .idea/ geliefert, welcher alle notwendigen Informationen für das Einbinden des Projekts in IntelliJ IDEA enthält. Dennoch ist es möglich, den Quellcode auch außerhalb dieser IDE zu kompilieren. Dazu müssen alle notwendigen Dateien in Bytecode, also .class Dateien transferiert werden und anschließend über die in 2.1 beschriebene Hauptklasse verbunden werden.

3. Herangehensweise

Diese Einheit soll die Herangehensweise an die Aufgabenstellung erläutern. Fundament für die Strategie bildet dabei das in der Vorlesung Wissensbasierte Systeme erworbene Wissen.

3.1 Datenimport

Die in der Aufgabenstellung genannten Daten werden in Form von drei .csv Dateien bereitgestellt. Für das CSV-Format ist kein aufwändiger Parsing Algorithmus notwendig. Es müssen zunächst die Reihen durch einen Zeilenumbruch separiert werden, anschließend werden die einzelnen Zellen durch einen anderen Separator (in diesem Fall ;) getrennt. Die Spalten werden im Kontext dieser Applikation als "Features" bezeichnet.

3.2 Diskretisierung

Um eine Emotion zu erkennen, werden zunächst die in der Vorlesung vorgestellten Methoden zur Erkennung von Gesichtselementen verwendet.

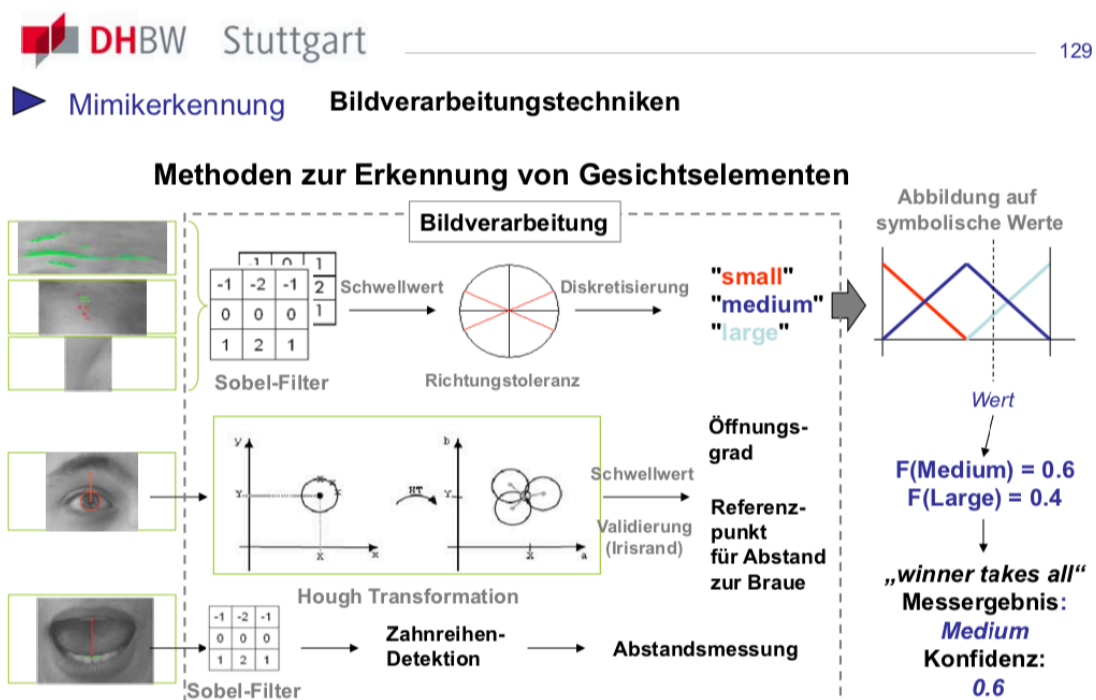


Abbildung 1: WBS-06-Evidenzen_2012.pdf Folie 129, Dirk Reichardt

Eine Messung des Bildes ist bereits in Form der Features aus den .csv Dateien vorhanden. Eine weitere Verarbeitung dieser Daten soll auf einem Gesichtsausdruck - folgend "Facial Expression" genannt - deuten. Diese Facial Expression soll mit einer gewissen Sicherheit, oder auch "Konfidenz", identifiziert werden. In den bereitgestellten Daten wurden die Facial Expressions Stirnfalten, Augenöffnung, Brauenabstand, horizontale Nasenfalten, vertikale Nasenfalten, Wangenfalten und Mundöffnung durch die Features *fob*, *lea*, *lbd*, *rea*, *rbd*, *hnc*, *vnc*, *lcw*, *rcw* und *ma* repräsentiert. Dabei geben die Features stetige, ganzzahlige, pixelbasierte Werte an. Die Facial Expressions hingegen werden diskreten Ausprägungsstufen zugeordnet. Diese beschränken sich auf *low*, *medium* und *high*. Da einige Facial Expressions wie z.B. Augenöffnung durch zwei Features auszeichnen, werden diese über den Durchschnitt beider zusammengeführt. Auch sind die pixelbasierten Werte abhängig von der Größe des Kopfes. So hat ein Mensch mit einem größeren Kopf tendenziell einen höheren Pixelwert für die Augenöffnung, als eine Person mit einem kleinen Kopf. Dementsprechend müssen die Features relativ zur Gesichtsgröße verarbeitet werden. Dazu werden die Metainformationen aus den Spalten *xright* und *ylo* verwendet. Folgende Formel demonstriert diese Vorgehensweise bei der Augenöffnung:

$$ea = \frac{(lea + rea) \cdot 10^4}{2 \cdot xright \cdot ylo}$$

Die durchschnittliche Augenöffnung wird errechnet und durch ein zehntausendstel der sich in der *Region of Interest* befindlichen Anzahl der Pixel geteilt. Dadurch entsteht ein zusammengeführter, normierter Wert. Dieser ist durch die Multiplikation mit 10000 effizienter in Programmiersprachen umzusetzen und einfacher für Menschen zu deuten. Für Features wie *ma*, bei denen keine zwei Werte vorhanden sind, entfällt die Durchschnittsberechnung. Die Normierung erfolgt jedoch trotzdem. Diese Werte müssen nun in die diskreten Ausprägungsstufen klassifiziert werden. Dazu werden die .csv Daten grafisch modelliert und von Hand analysiert. Abschließend werden drei Konfidenzfunktionen implementiert.

Sei x ein beliebiger zusammengeführter und normierter Feature-Wert, dann ist:

$$F_{low}(x) = -2a^{-2} \cdot x^2 + 1$$

$$F_{medium}(x) = -\frac{b}{d} \cdot |x - d| + 1$$

$$F_{high}(x) = \min(\frac{x}{2c}, 1)$$

Dabei gilt:

$$a = d - \frac{d}{2b}$$

$$b = var$$

$$c = d + \frac{d}{2b}$$

$$d = avg(x)$$

d bezeichnet den Durchschnittswert des Features innerhalb einer wohlbalancierten Menge von Frames. Im Kontext der Aufgabenstellung ist dies bei den ersten 35 Frames der Fall. b wird als Stellschraube eingesetzt, um für die jeweiligen Features die Steigung der Kurve anzupassen. Dies geschieht zum Schluss, im Optimierungsprozess.

Die Konfidenzwerte werden verglichen und der höchste wird als gegeben angenommen (vgl. Winner-Takes-All Prinzip). Das Resultat dieses Prozesses ist somit eine Aussage darüber, dass die Ausprägung der Stirnfalten eines Frames mit einer Konfidenz von 0,9 hoch ist.

3.3 Dempster-Shafer Regel

Das in 3.2 erlangte Resultat kann verwendet werden, um den in der Vorlesung erlernten Dempster-Shaefer Algorithmus mit Messungen anzureichern. In der Aufgabenstellung findet sich eine Auflistung der Facial Expressions, und welche Emotionen welche Ausprägungsstufen von ihnen üblicherweise mit sich bringen. Für die Implementierung von Dempsters Regel wird die in der Vorlesung bereitgestellte Ausführung von Ben Förnröhr verwendet. Als Messungen werden jeweils eine Menge an Emotionen übergeben, welche auf

die Ausprägung einer Facial Expression zutreffen. Zusätzlich wird die berechnete Konfidenz als statistische Wahrscheinlichkeit für die Menge an Emotionen übergeben. Wenn alle Facial Expressions als Messungen verarbeitet wurden, wird akkumuliert. Anschließend wird die Messung der höchsten Wahrscheinlichkeit analysiert. Es werden Plausibilität, Glaube und Zweifel ausgegeben. Dies soll für jedes Frame der Eingabedaten erfolgen.

4. Detaildokumentation

Die Datei wird im Laufe der Applikation durch drei verschiedene Abstraktionsebenen geführt. Diese Ebenen werden im Folgenden als "Model" bezeichnet.

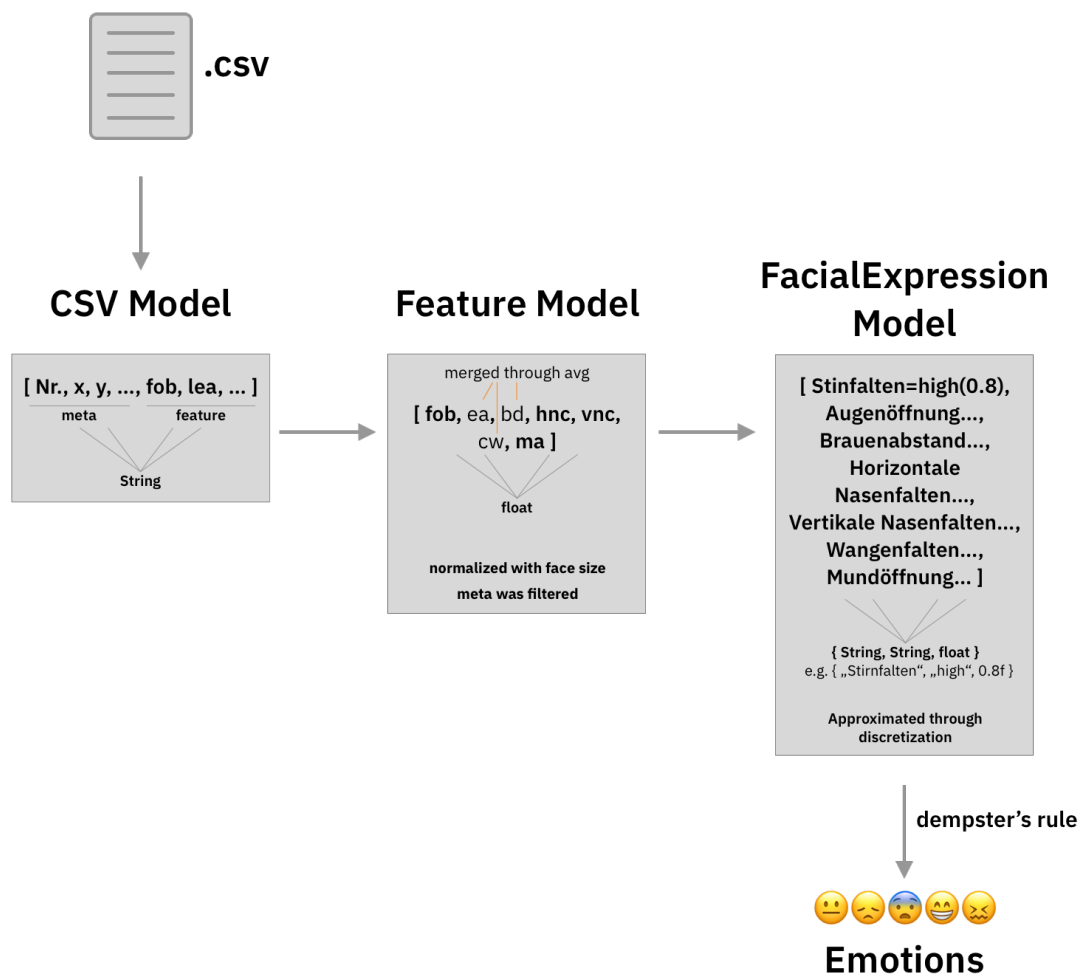


Abbildung 2: Selbstentwerfener Überblick über die abstrakte Applikationsarchitektur

4.1 CSV Model

Die auf der Festplatte vorhandene .csv Datei wird durch die CSVImporter Komponente in den Arbeitsspeicher geladen. Dort ist jede Reihe der .csv Datei als ein CSVFrame Objekt präsent. Dieses Objekt besteht aus einer Aufzählung der einzelnen Zellwerten des Frames. Es sind sowohl Feature Werte

als auch Metainformationen enthalten. Alle Werte sind durch Strings repräsentiert.

4.2 Feature Model

Im nächsten Schritt finden die in 3.2 beschriebenen Anpassungen statt, welche als *Data Cleansing* Prozess betrachtet werden können. Die Metainformationen werden gefiltert und sind nicht mehr auf dieser Ebene präsent. Zusammenhängende Features werden in einem Feature vereint, indem der Durchschnitt gebildet wird. Abschließend werden die Werte auf einen Wert normiert, welcher unabhängig von der Gesichtsgröße ist. Dabei werden die Daten in ein Fließkommaformat gebracht. Es wurde hierfür float verwendet, da Typkonvertierungen gegenüber der Dempster-Bibliothek vermieden werden sollten.

4.3 FacialExpression Model

Um Gesichtsausdrücke zu modellieren und der Dempster-Bibliothek zur Verfügung zu stellen, wird das Facial Expression Model eingeführt. Jedes der Werte aus dem Feature Model wird in die drei Konfidenzfunktionen eingesetzt (s. 3.2). Die Ausprägung mit der höchsten Konfidenz wird als zutreffend angenommen. Jede Ausprägung eines Gesichtsausdrucks passt auf eine Reihe von Emotionen. Diese werden als binäre Liste an die Bibliothek von Dempster übergeben.

5. Verifikation

Nach der Implementierung der Architektur kann diese verifiziert und optimiert werden. Die zu erkennenden Emotionen der ersten 35 Frames wurden teilweise durch die Aufgabenstellung definiert. Frames, für welche dort keine explizite Emotion vorgegeben wurde, werden von Hand analysiert und zugeordnet. So wird folgende Reihenfolge an Emotionen in den ersten 35 Frames über die drei Testdateien hinweg erkannt:

1;5		5 Sekunden	NEUTRAL
6;10		5 Sekunden	SADNESS
11;15		5 Sekunden	NEUTRAL
16;20		5 Sekunden	FEAR
21;25		5 Sekunden	NEUTRAL
26;30		5 Sekunden	JOY
31;35		5 Sekunden	NEUTRAL

Auf Basis dieser Daten ist es möglich, die erstellte Architektur zu testen und deren Performance zu messen. Indikator dafür ist der Prozentsatz der richtig klassifizierten Frames in diesem Bereich. Dieser wird jeweils für die drei Testdateien bereitgestellt. Der Test, über alle drei der bereitgestellten Dateien hinweg, kann durch Ausführen der alternativen Hauptklasse

```
de.emotreco.test.ClassifierTraining
```

durchgeführt werden.

Im nächsten Schritt wird eine Justierung der Variablen b für die einzelnen Features durchgeführt, um die Umwandlung von Feature Model zu Facial Expression Model zu optimieren. Dazu wird die beste Kombination von b Werten der verschiedenen Features empirisch ermittelt. Resultat ist eine Erkennungsrate von 88,57%.