

# Setup

1. Stellen Sie sicher, dass die Datei mit den zu analysierenden Daten im .csv Format vorhanden ist.
2. Platzieren Sie die Daten-Datei in dem Dataset-Ordner (in dem Projektordner).
3. Installieren Sie Python 3.7.
4. Öffnen Sie die Command Line und navigieren Sie zu dem Projekt-Ordner.
5. `$ python __main.py__`
6. Folgen Sie den Anweisungen des Programs.
7. Sie finden den Output in der Datei output.txt.

## Aufgabenstellung

Es wird eine CSV-Datei bereitgestellt, welche aus einem Video Gesichts-Charakteristika (bzw. Features) extrahiert. In den ersten vier Feature-Spalten wird die Position des Frames sowie die Größe des Frames angegeben. In den darauffolgenden Spalten werden in folgender Reihenfolge tatsächliche Features im Gesicht des Beobachteten dokumentiert:

- Stirnfalte
- Augenöffnung und Brauenabstand auf jeweils beiden Seiten
- horizontale und vertikale Falten auf der Nasenwurzel
- Faltenbildung auf beiden Wangenseiten
- Mundöffnung

Jede Zeile in der CSV-Datei stellt einen Frame des Videos dar. In der Aufgabenstellung ist eine bestimmte Reihenfolge bzw. Abfolge von Emotionen festgelegt.

Die Aufgabe ist es nun, für jeden Frame der Videosequenz (dem Programm im Format einer CSV-Datei bereitgestellt) mithilfe der ‚Dempster-Shafer-Regel‘ eine Emotion auszugeben, basierend auf den eingelesenen Features. Zwar sollen erst die Emotionen der letzten 25 Frames ausgegeben werden, da der Gruppe aber unbekannt ist wie die Testdaten bei der Bewertung aussehen, wird für jeden Frame die erkannte Emotion ausgegeben. Zusätzlich ist zu beachten, dass bei den Testdaten der Bewertung unvollständige Featurewerte auftreten können.

## Theorie

### Mapping: Aufgabenstellung auf Dempster-Shafer-Regel

In dieser Bearbeitung existiert eine Menge von Alternativen  $\Omega$ , die Menge aller möglichen Emotionen. Die Alternativenmenge ist durch die Aufgabenstellung vollständig gegeben und es ist anzunehmen, dass diese sich gegenseitig ausschließen. Somit sind alle Voraussetzungen für die Evidenztheorie erfüllt.

Gehen wir nun für die Emotionsbestimmung von einem Frame aus. Jedes gemessene Feature (ab Spalte 5) wird zu einem Basismaß. Dieses Basismaß besteht immer aus mindestens 2 Einträgen. Ein Eintrag setzt sich zusammen aus einer Menge an Emotionen, auch genannt ‚fokale Menge‘ und deren Konfidenz. Jede Evidenz hat immer ein  $\Omega$  - Element, also einen Eintrag, in der die fokale Menge die gesamte Menge von Alternativen ist.

Schlussendlich werden alle Basismaße, also alle gemessenen Features, akkumuliert. Daraus entsteht das Basismaß  $m_{\text{gesamt}}$ .

Aus den Basismaßen in  $m_{\text{gesamt}}$  kann dann für jede Emotion die Plausibilität errechnet werden. Die Emotion mit der höchsten Plausibilität wird schlussendlich als erkannte Emotion ausgegeben.

## Mapping: Datenaufbereitung

Die Daten, die in der CSV-Datei gegeben werden, können nicht genauso für den Algorithmus übernommen werden, da sie allein nicht für Aussagen über mögliche Emotionen ausreichen. In der Aufgabenstellung ist wörtlich beschrieben, welche Merkmale mit welcher Ausprägung für welche Emotion sprechen. Beispielsweise sind bei Traurigkeit eine hohe Zahl an Stirnfalten und eine große Augenöffnung festzustellen. Um diese Zuordnung vornehmen zu können, müssen die Zahlen also in Kategorien groß, mittel und klein (engl. large, medium, small) eingeordnet werden. Ein erster Ansatz wäre, manuell Grenzen für die Kategorien festzulegen. Dies funktioniert jedoch nur bei einer Datei, da sich die Zahlenbereiche in den eingegebenen Dateien gravierend unterscheiden können. Die Zahlen müssen also normalisiert werden.

Bei genauerer Untersuchung der Zahlen konnte festgestellt werden, dass sich nicht nur die Messwerte, sondern auch die Position und Größe der aufgenommenen Frames unterschieden. Bei der Umsetzung werden die Messwerte nun durch die Höhe des jeweiligen Frames (Spalte „ylo“) geteilt. Auf diese Weise erhält man über sämtliche Dateien hinweg Zahlen in den gleichen Zahlenbereichen. Da es sich nach der Normalisierung zum Teil um sehr kleine Werte handelt, werden einige Spalten mit Faktoren multipliziert, um die Auswertung zu vereinfachen. Anhand dieser neuen Werte können nun Grenzwerte für die einzelnen Gesichtsmerkmale und ihre Ausprägungen festgelegt werden. In allen Spalten konnte jeweils drei Wertebereiche gefunden werden, die genügend großen Abstand zueinander haben, um sie als Kategorien zu erkennen.

Im nächsten Schritt werden mögliche Emotionen für einen Frame anhand der Größen der Gesichtsmerkmale in diesem ausgewählt. Dazu wurde der Text aus der Aufgabenstellung auf eine Datenstruktur übertragen und alle Emotionen, auf die die Merkmale passen aufgenommen.

Der letzte Schritt beinhaltet die Ermittlung der Konfidenz  $K$  zu diesen fokalen Mengen  $M$ . Die Konfidenz ist ein Maß für die Zuverlässigkeit der Messung. Aus der Aufgabenstellung geht nicht hervor, unter welchen Umständen die Messung vorgenommen wurde und auch nicht, wie die Zuverlässigkeit bewertet wird.

Aus diesem Grund nehmen wir an, dass die Wahrscheinlichkeit dafür, dass eine Menge an Emotionen durch dieses Merkmal ausgewählt wird, die Konfidenz in angemessener Weise beschreiben kann. Sie wird daher in der Lösung berechnet mit:  $K = \frac{|M|}{|\Omega|}$ . Ein weiterer Ansatz war, die Messgenauigkeit mit einzubeziehen, indem die Wahrscheinlichkeit, dass eine Emotion auch mit dem Merkmal in einer anderen Größe auftreten kann, einberechnet wurde.

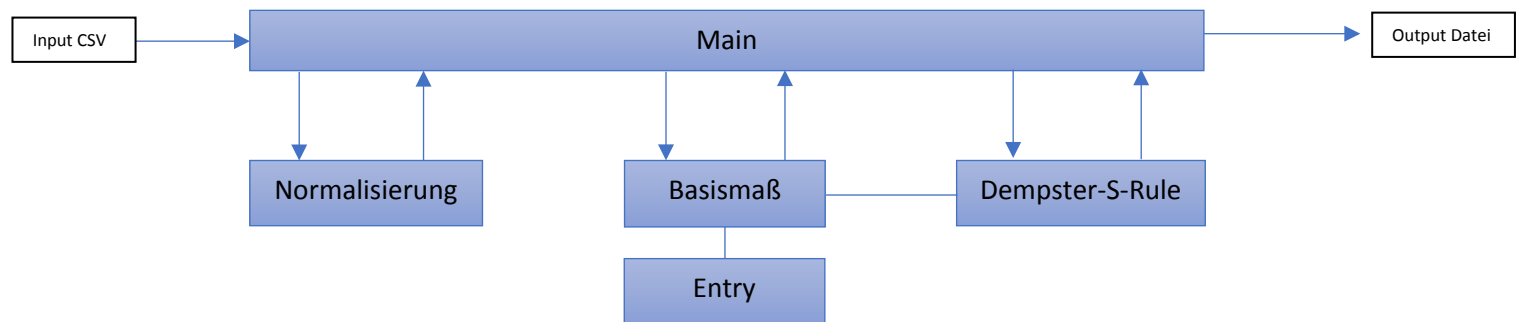
Formalisiert:  $K = \prod_{i=1}^{n=|M|} \text{Zahl der potenziellen Größen für } M_i \text{ bei diesem Merkmal} / 3$

Nach Umsetzung dieser Kalkulation verschlechterten sich jedoch die Ergebnisse. Besonders auffällig waren die ersten 5 Frames jeder Datei. Diese sollten laut Aufgabenstellung stets den Gesichtsausdruck „neutral“ darstellen. Bei der ersten Berechnung für die Konfidenz konnte dieses Ergebnis bestätigt werden, während mit dem zweiten Ansatz mit sehr hoher Plausibilität eine andere Emotion erkannt wurde. Daher ist in der finalen Umsetzung der simple Ansatz zu finden.

## Methode

### Umsetzung Algorithmus

Gesteuert wird der gesamte Ablauf des Programms von der Main-Klasse aus.



Der Ablauf ist in der Grafik dargestellt. Es wird in der Main der Input in Form einer CSV Datei eingelesen. Der Inhalt dieser wird dann in einer Schleife pro Zeile/Frame und zur Normalisierung weitergegeben. Die Normalisierung ist ein langer Prozess und deshalb im nächsten Subkapitel beschrieben.

Zurück kommt für den aktuellen Frame eine Liste; für jedes Feature im Input wird die Fokalmenge sowie die Konfidenz für diese Fokalmenge zurückgegeben. Im nächsten Schritt wird eine Liste von ‚Basis-Basismaßen‘ angelegt, es wird also für jedes Feature ein Basismaß angelegt. Diese Basismaßklasse greift im Konstruktor auf die Klasse Entry zu. Es wird zuerst ein Eintrag für die gegebene Fokalmenge und Konfidenz angelegt und zu der Entry-List dieser Evidenz hinzugefügt. Folgend daraus wird ebenfalls der Omega-Eintrag angelegt mit der Gegenwahrscheinlichkeit bzw. -Konfidenz.

Im nächsten Schritt wird durch diese ‚Basis-Basismaße‘-Liste iteriert und sie werden Schritt für Schritt akkumuliert. Dazu werden jeweils immer zwei Basismaße an die Dempster-Rule Klasse übergeben. Diese greift wiederum noch einmal auf die Basismaßklasse zu, um als Output auch ein Basismaß zurückgeben zu können. Dort wird dann tabellenartig jeder Eintrag miteinander verrechnet. Gegeben seien dazu die Basismaße  $m_k$  und  $m_j$ , die akkumuliert werden sollen zum Output-Basismaß  $m_o$ . Die DSR-Klasse unterscheidet dazu folgende Fälle:

1. Eines der Einträge ist Omega:  
Es wird in  $m_o$  ein Eintrag angelegt mit der Konfidenz bestimmt durch das Produkt der

Konfidenzen von  $m_k$  und  $m_j$  und der fokalen Menge gleich der fokalen Menge des Elements das nicht Omega ist.

2. Beide Einträge sind Omega:

Es wird in dem Output-Basismaß ein Eintrag angelegt, mit der Konfidenz bestimmt durch das Produkt der beiden Input-Einträge und der fokalen Menge Omega

3. Keines der Einträge ist Omega:

Hier werden zuerst die beiden fokalen Mengen miteinander verglichen und die Schnittmenge betrachtet

a. Die Schnittmenge ist leer

Es wird trotzdem ein Eintrag in dem Output-Basismaß mit entsprechender Wahrscheinlichkeit angelegt, dies wird später separat bearbeitet

b. Die Schnittmenge ist nicht leer

Es wird in dem Output-Basismaß ein Eintrag angelegt mit der Konfidenz bestimmt durch das Produkt der beiden Input-Einträge und der fokalen Menge gleich der bestimmten Schnittmenge

In jedem dieser Schritte wird vor dem Erstellen des Eintrags sicher gestellt, dass ein Eintrag mit derselben fokalen Menge nicht schon in dem Output-Basismaß existiert. Ist dies der Fall, wird die ausgerechnete Konfidenz des aktuellen Eintrags der Konfidenz des bereits existierenden Eintrags hinzuaddiert. Die einzige Ausnahme, in der dieser Vergleich nicht durchgeführt wird ist der Fall, in dem beide Input-Einträge Omega sind, da dieser Fall per Definition nur einmal pro Akkumulation vorkommt.

Als letzter Schritt bevor das Output-Basismaß zurückgegeben werden kann, werden die Einträge mit leeren fokalen Mengen bereinigt. Dazu wird die Konfidenz dieses Eintrags bestimmt und dann anhand des Faktors  $F_{\text{korrektur}} = 1 \frac{1}{1-k}$  die Konfidenzen der restlichen Einträge neu gesetzt.

Es ist zu beachten, dass nach der Neuberechnung nochmal ein Kontrollcheck implementiert wurde, der überprüft, ob die Konfidenzen addiert noch immer 1 ergeben.

Bei dem Kontrollcheck wie auch bei den Tests, die zur Kontrolle der Funktionen implementiert wurden, mussten wir leider auf Runden zurückgreifen. Da die Konfidenzen als float implementiert wurden, kam es zu Rundungsfehlern bei dem Ergebnis. Wir haben uns deshalb für den Check und die Tests dafür entschieden, die zu vergleichenden Ergebnisse auf 5 Nachkommastellen zu runden.

Wurde das Basismaß  $m_{\text{gesamt}}$  aus allen vorhergehenden Basismaßen berechnet, wird auf dieses Basismaß die Funktion `calculate_plausibility()` aufgerufen. Diese gibt eine Liste aller vorhandenen Emotionen in  $m_{\text{gesamt}}$  aus, sowie ihre Plausibilität. In der Main Klasse wird dann die Emotion mit der höchsten Plausibilität als Output gewählt und zusammen mit ihrer Plausibilität in die Output-Datei geschrieben.

## Ergebnisse

Die Output-Dateien des gegebenen Datasets sind im Dataset-Ordner bereit gestellt.

### emo\_muster\_2\_1.csv:

In der Aufgabenstellung ist eine konkrete Abfolge beschrieben: die ersten 5 Frames zeigen die Emotion ‚neutral‘, gefolgt von 3x jeweils 5 Frames einer unbekannten Emotion und danach 5 Frames der Emotion ‚neutral‘. Danach folgt eine beliebige Abfolge von Emotionen. Zum leichteren Verständlichkeit wurden die Gruppierungen mit Namen eingeteilt (siehe Grafik rechts).

Die Gruppierungen Neutral 1-4 wurden in diesem Dataset problemlos erkannt. Es gab keine Fehler oder Outlier und die Plausibilität von circa 68% schätzen wir als gut ein.

Die Gruppierung Unbekannt 1 wurde in diesem Dataset uniform und mit einer sehr hohen Plausibilität von 0.98 als ‚disgust‘ erkannt. Es ist also davon auszugehen, dass diese Analyse korrekt ist.

Die Gruppierung Unbekannt 2 wurde weitestgehend als ‚joy‘ erkannt, mit einer Plausibilität schwankend zwischen 0.62 und 0.92. In dieser Gruppierung gab es einen Frame, in dem ‚fear‘ erkannt wurde. Da dort die Plausibilität aber nur 0.57 beträgt und nur dieser Frame anders erkannt wurde, ist davon auszugehen, dass diese Gruppe der Emotion ‚joy‘ zugeordnet werden kann.

Die Gruppierung Unbekannt 3 ergab für die 5 Frames die Reihenfolge ‚neutral‘, ‚joy‘, ‚joy‘, ‚joy‘, ‚neutral‘. Bei diesem Frame kann kein eindeutiges Fazit über die vorliegende Emotion gegeben werden, da die Emotion nicht uniform bestimmt und die überwiegende Emotion ‚joy‘ nur mit einer Plausibilität von 0.56 erkannt wurde.

Für die Frames in der Kategorie random wurde ein gutes Ergebnis erzielt, da die meisten Emotionen in Gruppierungen von zwei oder mehr Frames erkannt wurden. Betrachtet man den Use-Case halten wir es für sehr wahrscheinlich, dass Emotionen für mindestens 2 Frames hintereinander auftreten und sich nicht schnell ändern. Mit besonders hoher Plausibilität wurden die Emotionen ‚disgust‘ (0.98) und ‚sorrow‘ (0.95) erkannt.

1	neutral	neutral 1
2	neutral	
3	neutral	
4	neutral	
5	neutral	
6	unbekannt	unbekannt 1
7	unbekannt	
8	unbekannt	
9	unbekannt	
10	unbekannt	neutral 2
11	neutral	
12	neutral	
13	neutral	
14	neutral	
15	neutral	unbekannt 2
16	unbekannt	
17	unbekannt	
18	unbekannt	
19	unbekannt	
20	unbekannt	neutral 3
21	neutral	
22	neutral	
23	neutral	
24	neutral	
25	neutral	unbekannt 3
26	unbekannt	
27	unbekannt	
28	unbekannt	
29	unbekannt	
30	unbekannt	neutral 4
31	neutral	
32	neutral	
33	neutral	
34	neutral	
35	neutral	random
36	random	

### emo\_muster\_2\_2.csv:

Die Gruppierungen Neutral 1-4 wurden wir im ersten Dataset ebenfalls korrekt erkannt.

Wie in Dataset 1 wurde die Gruppierung Unbekannt 1 als ‚disgust‘ erkannt, mit der gleichen Plausibilität.

Ebenfalls gleich ist die Gruppierung Unbekannt 2, die wieder als ‚joy‘ erkannt wurde. Hier gibt es jedoch eine Verbesserung zu Dataset 1; es wurden in diesem Dataset uniform und für 5 Frames ‚joy‘ erkannt, dies auch mit einer hohen Plausibilität von 0.79.

Anders zu Dataset 1 wurde die Gruppierung Unbekannt 3 hier als ‚neutral‘ erkannt. Die Plausibilität ist jedoch nicht hoch (~0.42) und deshalb ist diese Analyse als nicht zuverlässig zu bewerten. Es gilt hier darauf zu verweisen, dass es auch schon in Dataset 1 Probleme gab, die Emotionen ‚joy‘ und ‚neutral‘ voneinander zu unterscheiden. Dieser Vergleich wird in der Gesamtbewertung noch einmal angesprochen.

In der Gruppierung Random tritt jede erkannte Emotion in Gruppierungen von mindestens 2 Frames auf, hier ist auf die Deutung der Gruppierung Random im vorherigen Dataset zu verweisen. Wieder

wurden die Emotionen ‚disgust‘ und ‚sorrow‘ mit sehr hoher Plausibilität erkannt. In diesem Dataset wurde auch die Emotion ‚joy‘ mit einer vergleichsweise hohen Plausibilität erkannt.

#### emo\_muster\_2\_3.csv:

Ähnliche Zahlen wie in Dataset 2, gleiche Analyse.

#### Gesamt:

Abschließend stellt man fest, dass die Emotionen ‚disgust‘ und ‚sorrow‘ durchweg mit sehr hoher Plausibilität erkannt werden. Im Gegensatz dazu hat der Algorithmus Probleme, die Emotionen ‚neutral‘ und ‚joy‘ voneinander zu unterscheiden. Das kann daran liegen, dass die den Emotionen zugeordneten Features sich sehr ähneln und nur in der Ausprägung unterscheiden.

Eine weitere möglich Fehlerquelle sehen wir in den unterschiedlichen Größen der Bounding Box. Wir haben beobachtet, dass mehr Fehler (vor allem in der Unterscheidung von ‚joy‘ und ‚neutral‘) auftreten, je kleiner die Bounding Box ist. Wir vermuten, dass die Komprimierung zu Ungenauigkeiten in den gemessenen Zahlen führt.

Wir vermuten schlussendlich, dass alle drei Datasets die gleiche Reihenfolge von Emotionen zeigen und sich nur in der Größe der Bounding Box unterscheiden.

## Future Work

Bei der Bearbeitung der Aufgabenstellung gab es in der Gruppe zwei Vorschläge für mögliche Erweiterungen, die aufgrund der Zeit leider nicht mehr eingebunden werden konnten. Wir möchten dies aber trotzdem nennen, da wir denken, dass diese Erweiterungen die Analyse nochmals verbessern könnten.

#### Kontext bei real-world Anwendung

Für diese Bearbeitung haben wir uns dazu entschieden, für jeden Frame die erkannte Emotion auszugeben. Laut der Aufgabenstellung ändert sich (zumindest am Anfang) in bestimmten Abständen die Emotion. Nun ist es nicht unwahrscheinlich, dass die Emotion nicht durchgehend klar zu erkennen ist oder die Testperson beim Wechseln des Gesichtsausdrucks keine ‚klare‘ Emotion zeigen kann. In einem realen Anwendungsszenario wäre es also sinnvoll – wenn man die Abfolge und Länge der Emotionen kennt – die Analyse eines Frames mit Kontext zu betrachten. Folgendes Outputbeispiel verdeutlicht dies (es ist bekannt, dass eine Emotion für 5 Sekunden gehalten wird):

Fear – 0.97	Joy – 0.82
Fear – 0.97	Joy – 0.82
Fear – 0.97	Joy – 0.82
Fear – 0.97	Sadness – 0.60
Neutral – 0.56	Joy – 0.82

Hier ist deutlich zu sehen, dass man leicht Fehlinterpretationen eliminieren kann, in dem man bei einem oder zwei Outliern diese ‚ignoriert‘ bzw. nur eine Emotion pro 5 Sekunden ausgibt.

## K-Means Clustering

Die aus der Datenverarbeitung gewonnenen Werte können in der Anwendung so zur Berechnung genutzt werden. Dennoch wurden die Intervalle für die Kategorisierung nach Größe/ Ausprägung des Merkmals per Hand festgelegt. Bei der Nutzung einer solchen Anwendung in der Welt sind diese Werte womöglich je nach Person unterschiedlich und die Intervalle unterschiedlich groß. Aus diesem Grund wäre ein Ansatz, die Wertebereiche für die Größen festzulegen, auf alle Zahlen einer Spalte in der CSV-Datei einen K-Means-Clustering-Algorithmus laufen zu lassen. Dieser soll die Zahlen in 3 Cluster unterteilen. Dasjenige Cluster, das die größten Zahlen enthält, entspricht den Zahlen für Kategorie „l“, das mit den kleinsten entspricht „s“ und das mittlere der Kategorie „m“. Auf diese Weise können die Intervalle für jeden Datensatz und jede Person individuell erstellt werden.