DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Automatisierte Analyse von Sprachsignalen unter Verwendung der Fourier-Analyse

Projektarbeit

in der fünften Praxisphase

an der Fakultät Technik
im Studiengang Informationstechnik

an der DHBW Ravensburg
Campus Friedrichshafen

von
Johannes Brandenburger

17.04.2023

Bearbeitungszeitraum:	09.01.2023 - 17.04.2023
Matrikelnummer, Kurs:	1688304, Informationstechnik (B. Sc.) TIT20
Dualer Partner:	Geberit Verwaltungs GmbH
Betreuer im Partnerunternehmen:	Moritz Kaltenstadler

Selbstständigkeitserklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017.

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema:

Automatisierte Analyse von Sprachsignalen unter Verwendung der Fourier-Analyse

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Pfullendorf, 17.04.2023

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Listingsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Projektumfeld

Diese Projektarbeit findet im Rahmen der Studienarbeit „Nutzerauthentifizierung anhand Stimm-Analyse“ von Henry Schuler, Lukas Braun und Johannes Brandenburger statt. Ziel der Studienarbeit ist es, die Frage zu beantworten, ob eine Benutzerauthentifizierung anhand Stimmanalyse sicher, effektiv und praktikabel ist. Dazu werden zunächst Versuche mit verschiedenen Analyse-Methoden durchgeführt und auf dessen Basis eine Applikation entwickelt, die eine Stimm-Authentifizierung demonstriert.

Eine dieser Analysemethoden ist die Fourier-Analyse, die das Thema dieser Projektarbeit ist. Die Ergebnisse dieser Projektarbeit werden in der Studienarbeit genutzt und Teile dieser schriftlichen Ausarbeitung sind in der Studienarbeit wiederzufinden.

1.2 Problemstellung

Für eine Benutzerauthentifizierung anhand der menschlichen Stimme müssen zunächst Eigenschaften aus einem Sprachsignal gefiltert werden, die als Datenbasis für ein Modell, wie beispielsweise ein Neuronales Netz, dienen können.

1.3 Ziel

Ziel der Arbeit ist die Entwicklung eines Programms, das mithilfe der Fourier-Analyse weiterverwendbare Merkmale aus einem Sprachsignal extrahiert. Anschließend soll bewertet werden, ob ein relevanter Zusammenhang zwischen den Ergebnissen und der Herkunft der Stimmsignale besteht.

2 Theoretische Grundlagen

2.1 Sprachauthentifizierung

> *„Authentifizierung ist das Prüfen einer Behauptung über eine partielle Identität. Nach einer erfolgreichen Authentifizierung kann die partielle Identität einer Rolle zugeordnet werden.“* (Hühnlein 2008, S. 162)

Einfacher erklärt, ist Authentifizierung das Überprüfen, ob jemand wirklich die Person ist, die er vorgibt zu sein. Nach erfolgreicher Überprüfung werden dieser Person bestimmte Berechtigungen zugewiesen.

Menschen betreiben heutzutage Online-Banking, laden ihre kompletten privaten Daten in Speicherungsmedien in der Cloud und entsperren ihr Smartphone etliche Male pro Tag. Die Wichtigkeit und Menge an Authentifizierungsvorgängen erfordern ein immer höheres Maß an Sicherheit und Bequemlichkeit bei den Authentifizierungsverfahren. In den letzten Jahren sind immer wieder neue Arten von Authentifizierungsmethoden erschienen, wie z.B. den Fingerabdruckscanner, die Iriserkennung oder die Gesichtserkennung.

Bisher ist die Sprachauthentifizierung noch nicht weit verbreitet, was vor allem daran liegt, dass die Methode in vielen Situationen nicht praktikabel ist. Allerdings gibt es durchaus Anwendungsfälle, in denen die Sprachauthentifizierung eine gute Alternative zu anderen Authentifizierungsmethoden darstellt. Beispiele hierfür wären die Authentifizierung von Menschen mit Behinderungen und die Authentifizierung in Telefongesprächen.

2.2 Fourier Analyse

Jean-Baptiste Joseph Fourier (1768 - 1830) war ein französischer Mathematiker. Er war in seinem Mathematikstudium Schüler von Lagrange, Laplace und Monge und veröffentlichte 1822 seine Arbeit „Theorie der Analytischen Funktionen“. Damit legte er die Grundlagen für die Fourier-Reihe und die Fourier-Transformation. Diese Verfahren sind bis heute in der Mathematik und Physik von großer Bedeutung und bilden die Grundlage für die Fourier-Analyse. Fourier gilt daher als einer der wichtigsten Mathematiker aller Zeiten. (vgl. Heinz Klaus Strick 2012, o. S.)

Der Begriff Fourier-Analyse ist in der Literatur hauptsächlich für zwei Dinge verwendet worden. Zum einen würdigt er die Rolle Fouriers in der Entwicklung der Mathematik.

Zum anderen ist er ein Überbegriff für eine Reihe an mathematischen Verfahren, die von Fourier entwickelt wurden. (vgl. Picard 1996, S. 1)

2.2.1 Fourier-Reihe

Die Fourier-Reihe ist ein Verfahren, mit dem eine periodische Funktion in eine unendliche Summe von Sinus- und Cosinusfunktionen zerlegt werden kann. Wenn ein Signal $s(t)$ folgenende Eigenschaften aufweist:

- $s(t)$ ist periodisch mit der Periode T (d.h. $s(t+T) = s(t)$)
- $s(t)$ ist stetig mit endlich vielen Sprungstellen im Intervall $[0, T]$
- $\int_0^\infty |s(t)|dt < \infty$

dann kann $s(t)$ in

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \cdot sin(n\omega_0 t) \tag{1}$$

zerlegt werden, wobei

$$a_0 = \frac{2}{T} \int_0^T s(t)dt \tag{2}$$

$$a_n = \frac{2}{T} \int_0^T s(t) \cdot cos(n\omega_0 t)dt \tag{3}$$

$$b_n = \frac{2}{T} \int_0^T s(t) \cdot sin(n\omega_0 t)dt \tag{4}$$

$$\omega_0 = \frac{2\pi}{T} \tag{5}$$

sind. (vgl. Rieß und Wallraff 2018, S. 19f)

Die Fourier-Koeffizienten a_n und b_n sagen also aus, wie stark einzelne Frequenzen in $s(t)$ vorkommen. Diese Frequenzanteile sind als Output der Fourier-Reihe zu verstehen und können als Merkmale für periodische Signale verwendet werden. a_0 ist der Mittelwert der Funktion $s(t)$.

2.2.2 Kontinuierliche Fourier-Transformation

Anders als bei der Fourier Reihe, können mit der verallgemeinerten kontinuierlichen Fourier-Transformation auch aperiodische Signale zerlegt werden, die folgende Eigenschaften aufweisen:

- $s(t)$ ist stückweise stetig
- $\int_{-\infty}^{\infty} |s(t)|dt < \infty$

Die Fourier-Transformation liefert eine Transformierte Funktion $\underline{S}(f)$, die als Spektrum des Signals $s(t)$ bezeichnet wird. Es wird also die Dimension Zeit t in die Dimension Frequenz f transformiert. Die Formel für die Fourier-Transformation lautet:

$$\underline{S}(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j2\pi ft} dt \tag{6}$$

Wie bereits an der Formel zu erkennen ist, ist das Ergebnis der Fourier-Transformation eine komplexe Funktion mit einem Realteil $\Re(\underline{S}(f))$ und einem Imaginärteil $\Im(\underline{S}(f))$.

Da bei der Weiterverarbeitung in der Regel nur der Normalteil des Spektrums benötigt wird, wird die Fourier-Transformation in der Praxis meistens in der Form

$$\Re(\underline{S}(f)) = \int_{-\infty}^{\infty} s(t) \cdot cos(2\pi ft) dt \tag{7}$$

durchgeführt. (vgl. Picard 1996, S. 350f)

2.2.3 Diskrete Fourier-Transformation

Bei den bisherigen Analyseverfahren, wurden immer kontinuierliche Signale betrachtet. Das heißt, dass für jeden Zeitpunkt t ein Wert $s(t)$ existiert.

Bei der Diskreten Fourier-Transformation (DFT) können diskrete Signale verarbeitet werden, die aus einer Folge von N Werten $s_0, s_1, \ldots, s_{N-1}$ bestehen, weshalb diese Methode besonders interessant für elektronisch aufgezeichnete Signale ist.

Da keine kontinuierliche Funktion vorliegt, besteht keine Möglichkeit mehr, ein Integral zu bilden, weshalb die Formel jetzt mit einer Summe aufgestellt ist:

$$S_k = \frac{1}{N} \sum_{n=0}^{N-1} s_n \cdot e^{-j2\pi \frac{kn}{N}} \tag{8}$$

Der Output der DFT ist wieder eine komplexe Wertereihe, deren Realteil $\Re(S_k)$ und Imaginärteil $\Im(S_k)$ die Amplitude und Phase der k-ten Frequenzkomponente des Signals s_n beschreiben. Der Realteil lässt sich durch die Umformung der Formel in die Form

$$\Re(S_k) = \frac{2}{N} \sum_{n=0}^{N-1} s_n \cdot cos(2\pi \frac{kn}{N}) \tag{9}$$

berechnen. (vgl. Smith 1997, S. 567ff)

Um auf die jeweiligen Frequenzanteile des Signals zu schließen, können die Output-Bins S_k mit folgender Formel in die Frequenzen f_k umgerechnet werden:

$$f_k = \frac{k}{N} \cdot f_s \tag{10}$$

wobei f_s die Abtastrate des Signals ist.

Fast Fourier Transformation Die Diskrete Fourier-Transformation würde sich nach der obigen Formel implementieren und für jedes Signal nutzen lassen. Allerdings ist die Komplexität der DFT mit $\mathcal{O}(N^2)$ sehr hoch, weshalb die Fast Fourier Transformation (FFT) entwickelt wurde, die den gleichen Output liefert, aber mit einer Komplexität von $\mathcal{O}(Nlog(N))$. (vgl. Beucher 2011, S. 338)

Die Fast Fourier Transformation ist also eine für Computer optimierte Implementierung der Diskreten Fourier-Transformation. Eine genaue Erklärung, warum FFT signifikant weniger Berechnungen benötigt, würde für diese Projektarbeit zu sehr ins Detail gehen. Einfach gesagt, nutzt der Algorithmus aber die Eigenschaft von Sinus- und Cosinus-Wellen, dass Wellen mehrfacher Frequenz an bestimmten Stellen dieselben Werte annehmen und so nur einmal berechnet werden müssen.

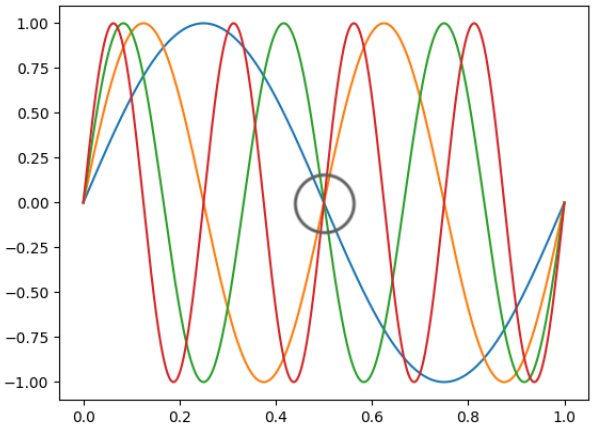

Abbildung 2.1: Sinuswellen mit 1hz, 2hz, 3hz, 4hz überlagern sich bei $t = 0,5$
Quelle: Eigene Darstellung

Durch diese Eigenschaft und das rekursive Aufteilen des Signals in gerade und ungerade Datenpunkte spart der Algorithmus sehr viele Berechnungen ein. (vgl. Oppenheim, Schafer und Buck 1999, S. 643)

2.3 MFCC

Die diskrete Fourieranalyse ist durch das Herausfiltern der vorhandenen Frequenzen bereits gut geeignet, um wesentliche Merkmale, wie die Stimmlage und Stimmfarbe einer Aufzeichnung, zu bestimmen. Allerdings werden in der Praxis meistens die Mel Frequency Cepstral Coefficients (MFCC) benutzt, die zwar auf einer Fourier-Analyse basieren, aber noch weitere Berechnungsschritte erfordern.

Der Vorteil von Merkmalen wie MFCC ist zunächst, dass so Audio-Signale mit verschiedenen Längen und Abtastraten besser miteinander verglichen werden können. Das liegt daran dass, bei einer Analyse der MFCC eine feste Anzahl Koeffizienten entsteht. Die Anzahl hängt dabei, anders als bei einer Fouriertransformation, nicht von der Frame-Länge und Abtastrate ab. Zudem sind die MFCCs mehr auf die menschliche Wahrnehmung von Tönen abgestimmt und robuster gegenüber Hintergrundrauschen.

Um die Mel Frequency Cepstral Coefficients zu berechnen sind mehrere Schritte nötig:

1. **Framing**
   Unterteilen des Signals 20 - 40 ms Abschnitte
2. **Diskrete Fourier-Transformation**
   Bestimmung der Frequenzkomponenten
3. **Logarithmierung**
   Näher an menschlicher Wahrnehmung
4. **Abbildung auf die Mel-Skala**[1]
   Zusammenfassung von Frequenzen zu Mel-Bänder mithilfe von Dreiecksfiltern
5. **Diskrete Kosinustransformation**
   Ähnlich zur inversen Fourier-Transformation

(vgl. Logan 2000, S. 2)

[1] Das Mel ist eine Maßeinheit für die wahrgenommene Tonhöhe. Das Mel verläuft ab einer gewissen Frequenz logarithmisch und nicht mehr proportional zu Hertz, da ein Mensch die Tonhöhe logarithmisch wahrnimmt.

3 Technische Umsetzung einer automatischen Analyse

3.1 Konzept

Um Eigenschaften aus einem Audiosignal zu gewinnen, soll ein Python-Programm erstellt werden, das folgende Anforderungen umsetzt:

1. **Einlesen einer Audiodatei** (sodass dass Signal vorliegt)
2. **Entfernen von Hintergrundrauschen**
3. **Entfernen von Pausen** (Stille)
4. **Unterteilung in Frames**
5. **Berechnung von MFCCs, Delta-MFCCs und Delta-Delta-MFCCs**

Zudem soll eine Klasse erstellt werden, die die generierten Merkmale auf ihre Relevanz überprüft. Dies soll zunächst durch einfache mathematische Vergleiche geschehen. Wenn die Vergleiche zu keinen erkennbaren Zusammenhängen von verschiedenen Sprechern führt, müssen andere Auswertungsmethoden, wie ein Neuronales Netz dazu verwendet werden.

3.2 Implementierung

Für die Anforderungen 1 - 4 wird die eigenständige Klasse `AudioPreprocessor` erstellt, welche auch für die Vorbereitung für die Generierung von anderen Merkmalen verwendet werden kann. Das Einlesen einer Audiodatei wird hierbei mit einer Funktion der Librosa-Bibliothek realisiert. Librosa ist ein bekanntes Python Paket, das für die Analyse von Musik und anderen Audiosignalen entwickelt wurde. Auch für das Entfernen von Hintergrundrauschen wird ein externer Algorithmus von Tim Sainburg verwendet. Das Entfernen von Pausen wird selber implementiert, wie im Listing 3.1 zu sehen ist. Hierbei werden Teile des Signals gelöscht, wenn sich die Amplitude über einen gewissen Zeitpunkt unter einem gewissen Schwellenwert befindet.

```
# iteriere über das Signal
for i, amp in enumerate(y):
  if abs(amp) < threshold:
    # wenn die Amplitude unter der Schwelle ist, starte die Zählung
    counter_below_threshold += 1
  else:
```

```
    # wenn die Amplitude über der Schwelle ist und über einen gewissen Zeitraum unter der
    Schwelle war...
    if counter_below_threshold > pause_length:
      # ...lösche die Samples, die unter der Schwelle waren (abzüglich eines Offset)
      for index in range(i-counter_below_threshold+keep_at_start_and_end, i-
    keep_at_start_and_end):
        indices_to_remove.append(index)
    counter_below_threshold = 0
```

Listing 3.1: Entfernen von Stille in einem Audiosignal

Anschließend wird das gesamte Signal in kurze überlappende Frames unterteilt und deren Anfang und Ende mit einer Windowing-Funktion geebnet.

Für die Berechnung der Merkmale (Anforderung 7) wird die Klasse `FeatureExtraction` erstellt, unter welcher später auch noch andere Merkmale wie LPC[2] hinzugefügt werden. Auch bei der Berechnung der Mel Frequency Cepstral Coefficients wird auf Librosa gesetzt, um Implementierungsfehler zu vermeiden und einen stärkstmöglich optimierten Algorithmus zu verwenden. Die Methode `extract_mfcc()` (siehe Listing 3.2) berechnet für eine Liste an Frames sowohl 13 MFCCs als auch die Ableitungen Delta-MFCCs und Delta-Delta-MFCCs. Insgesamt werden so also $13 \cdot 3 = 39$ Koeffizienten pro Frame berechnet.

```
def extract_mfcc(frames, sr):
  n_mfcc = 13 # Anzahl an Koeffizienten (typisch:  12-20)

  mfccs = []
  for frame in frames:
    # Berechnung von MFCCs für jeden Frame
    mfcc = librosa.feature.mfcc(y=frame, sr=sr, n_mfcc=n_mfcc)
    mfcc_of_frame = np.mean(mfcc.T, axis=0)
    mfccs.append(mfcc_of_frame)
  mfccs = np.array(mfccs)

  # Berechnung von Delta-MFCCs und Delta-Delta-MFCCs
  delta_mfccs = librosa.feature.delta(mfccs, order=1, mode='nearest')
  delta_delta_mfccs = librosa.feature.delta(mfccs, order=2, mode='nearest')
  features = np.concatenate((mfccs, delta_mfccs, delta_delta_mfccs), axis=1).tolist()
  return features, n_mfcc * 3
```

Listing 3.2: Generierung der MFCC Merkmale

[2]Linear Predictive Coding ist ein weiteres Analyseverfahren und Merkmal eines Audiosignals. Dieses Thema wird in einem anderen Teilprojekt der Studienarbeit behandelt.

3.3 Prüfung der Ergebnisse

Die eigentliche Implementierung der Audioanalyse ist dank den erwähnten Bibliotheken wie Librosa nicht sehr aufwändig. Allerdings sind die generierten Koeffizienten zunächst einmal sehr abstrakt und daher mit bloßem Auge nicht auf Relevanz zu bewerten. Die Nachvollziehbarkeit kann durch eine Visualisierung der Merkmale zwar verbessert werden, allerdings bleibt das Belegen von Zusammenhängen trotzdem schwierig. Deshalb wird für die Prüfung der Ergebnisse ein eigenes Programm entwickelt.

Zunächst wird ein Datenset benötigt, das Aufnahmen von mehreren Sprechern beinhaltet. Hierfür wird das „Speaker Recognition Audio Dataset“ von Vibhor Jain benutzt.

Um die Qualität der Merkmale zu bewerten, wird zunächst die Standartabweichung der Koeffizienten innerhalb eines Sprecher-Datensatz mit der Standartabweichung der Koeffizienten zwischen den verschiedenen Sprecher-Datensätzen verglichen. Dies liefert allerdings keine guten Ergebnisse, wie in der Ausgabe 3.3 zu sehen ist.

```
Average deviation of speakers:  8.7022
Deviation between speakers:     8.2959
```

Listing 3.3: Ausgabe der Standartabweichungen

Die Daten lassen darauf schließen, dass sich die Koeffizienten verschiedener Audiodateien nicht einfach linear über eine Abweichung vergleichen lassen. Aus diesem Grund wird für die weitere Bewertung ein Neuronales Netz entwickelt, das die Merkmale als Inputparameter hat.

Die Klasse `FeatureEvaluator` generiert zunächst mit den zuvor beschriebenen Klassen die Merkmale für 20.000 Frames pro Sprecher, um eine ausreichende Datenmenge sicherzustellen. Anschließend wird ein Neuronales Netz mit vier Layern und insgesamt 817 Neuronen erstellt und mit dem Großteil der Datenmenge (5/6) trainiert, wie im Listing 3.4 abgebildet ist. Für das Neuronale Netz wird das von Google entwickelte Machine-Lerning-Paket Tensorflow verwendet.

```
# Definition und Kompilierung des Neuronalen Netzes
model = tf.keras.Sequential([
  tf.keras.layers.Flatten(input_shape=[self.n_features*20]),
  tf.keras.layers.Dense(16, activation=tf.nn.relu),
  tf.keras.layers.Dense(16, activation=tf.nn.relu),
  tf.keras.layers.Dense(self.n_speaker, activation=tf.nn.softmax)
])
model.compile(optimizer=tf.optimizers.Adam(),
  loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Trainieren des Neuronalen Netzes
```

```
X, Y = self.unison_shuffled_copies(self.nn_data)
model.fit(X[int(5*self.total_input_chunks/6):],
          Y[int(5*self.total_input_chunks/6):],
          epochs=1000)
```

Listing 3.4: Neuronales Netz

Anschließend kann mit der Methode `evaluate_model_with_example(...)` eine bisher unbekannte Audiodatei in das Neuronale Netz geladen werden. Die Ausgabe indiziert dann, welchem Sprecher die Audiodatei von dem Neuronalen Netz zugeordnet werden kann. So kann entschieden werden, ob die Koeffizienten ausreichen, um ein neues Audiosignal richtig zu klassifizieren.

```
feature_evaluation.evaluate_model_with_example(speaker_index=2, file_index=12)

# Output:
Test accuracy: 0.8463385105133057
Speaker 1: 19
Speaker 2: 68
Speaker 3: 2
Speaker 4: 7
Speaker 5: 4
```

Listing 3.5: Evaluation der generierten Merkmale

An dem Ergebnis in Listing 3.5 ist zu sehen, dass die Audiodatei zu 68 % dem richtigen Sprecher zugeordnet wird. Das reicht noch nicht für eine komplett eindeutige Identifizierung eines Sprechers, allerdings ist zu sehen, dass die generierten MFCC relevante Koeffizienten bei der Audioanalyse sind.

4 Fazit

Die Fourier-Analyse und die dadurch resultierende Generierung der Mel Frequency Cepstral Coefficients ist ein wichtiger Schritt für eine Sprachauthentifizierung, wie an den Bewertungs-Ergebnissen zu erkennen ist: Die MFCCs repräsentieren die Stimmhöhe- und Farbe der Sprecher in einheitlichen Werten und lassen diese so miteinander vergleichen. Die Genauigkeit einer Klassifikation anhand der Koeffizienten kann durch ein optimierteres Netz und mehr Inputdaten noch wesentlich erhöht werden. Aber schon mit dem gezeigten einfachen Neuronalen Netz zeigen Stichproben, dass die Merkmale definiv die Stimme eines Sprechers repräsentieren können, womit das Ziel dieser Arbeit erreicht wurde.

Eine Herausforderung war ein geeignetes Datenset zu finden, das den Anforderungen entspricht. Zudem war die Überprüfung der berechneten Koeffizienten aufwändiger als geplant, da hierfür ein Neuronales Netz nötig war.

Für die Studienarbeit „Nutzerauthentifizierung anhand Stimm-Analyse“ sind noch weitere Merkmale, wie LPC und zeitanalytische Merkmale, nötig. Deren Generierung und das Entwickeln eines optimierten und sicheren Neuronalen Netzes sind die nächsten Schritte der Studienarbeit.

Literatur

Beucher, Ottmar (2011). *Signale und Systeme: Theorie, Simulation, Anwendung Eine beispielorientierte Einführung mit MATLAB*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 9783642202940.

Heinz Klaus Strick (2012). *Joseph Fourier (1768–1830)*. Spektrum: Der Mathematische Monatskalender. URL: https://www.spektrum.de/wissen/joseph-fourier-1768-1830/1156113 (besucht am 27. 01. 2023).

Hühnlein, Detlef (März 2008). „Identitätsmanagement: Eine visualisierte Begriffsbestimmung“. In: *Datenschutz und Datensicherheit - DuD* 32.3, S. 161–163. ISSN: 1614-0702, 1862-2607. DOI: 10.1007/s11623-008-0023-x. URL: http://link.springer.com/10.1007/s11623-008-0023-x (besucht am 24. 01. 2023).

Logan, Beth (2000). „Mel Frequency Cepstral Coefficients for Music Modeling“. In.

Oppenheim, Alan V., Ronald W. Schafer und John R. Buck (1999). *Discrete-time signal processing*. 2nd ed. Upper Saddle River, N.J: Prentice Hall. 870 S. ISBN: 9780137549207.

Picard, Rainer (1996). „Fourier-Analyse“. In: DOI: 10.25521/HQM27. URL: https://madoc.bib.uni-mannheim.de/50958/ (besucht am 27. 01. 2023).

Rieß, Bernhard und Christoph Wallraff (2018). „Fourier-Reihe“. In: *Übungsbuch Signale und Systeme*. Wiesbaden: Springer Fachmedien Wiesbaden, S. 19–32. ISBN: 9783658192259 9783658192266. DOI: 10.1007/978-3-658-19226-6_2. URL: http://link.springer.com/10.1007/978-3-658-19226-6_2 (besucht am 07. 02. 2023).

Smith, Steven W. (1997). *The scientist and engineer's guide to digital signal processing*. 1st ed. San Diego, Calif: California Technical Pub. 626 S. ISBN: 9780966017632.