Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Prof. Dr.-Ing. André Kaup

Bericht des Forschungspraktikums

Zuverlässige Bestimmung von spezifischen Energiewerten für die Videodecodierung

von Marco Bader

August 2018

Betreuer: Christian Herglotz

Kurzfassung

In diesem Forschungspraktikum wurden die spezifischen Energiewerte bei der Videodecodierung hinsichtlich ihrer Bestimmung untersucht. Bei der Videodecodierung wird ein Bistrom erhalten, welcher die Informationen eines Videos enthält. Dieser Bitstrom wird dann decodiert, so dass das Video reproduziert wird. Dabei werden verschiedene Operationen unterschiedlich oft ausgeführt. Das Energiemodell, das die benötigte Decodierenergie schätzt, hängt zum einen von der Auftrittsthäufigkeit der Operationen und den spezifischen Energiewerten der einzelenen Operationen ab. Diese spezifischen Energiewerte bestimmen wie viel Energie beim einmaligen ausführen der Operation verwendet wird. Für die Bestimmung der spezifische Energiewerte werden verschiedene Trainingssets, Modelle und verschiedene Methoden der Kreuzvalidierung verwendet. Das Ergebnis der Arbeit ist, dass ein Überblick der verschiedenen verwendeten Kombinationen gegeben wird und dabei zu erkennen ist, welche Kombinationen für die Bestimmung der spezifischen Energiewerte sich am besten eignen. Des Weiteren wird gezeigt welche Operationen in der Videodecodierung sich zuverlässig bestimmen lassen und welche nicht. Zuletzt wurden neue Trainingssets, welche hinsichtlich der in dieser Arbeit erzielten Ergebnisse in der Bestimmung der spezifischen Energiewerte erstellt.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

In den letzten Jahren hat die Nutzung von mobilen Multimediageräten, wie beispielsweise Smartphones, stark zugenommen, sodass sie heutzutage im Alltag etabliert sind. So waren es im Dezember 2010 noch 14,03 Millionen Nutzer in Deutschland und im Jahre 2017 bereits 54 Millionen [Sta17]. Auf diesen mobilen Endgeräten werden auch regelmäßig Streaming-Dienste, wie YouTube, Netflix, Amazon Prime Video und Weitere benutzt. Allerdings steigert das Streamen oder Abspielen von Videos auf den Geräten den Akkuverbrauch. Um den Akkuverbrauch beim Abspielen der Videos möglichst gering zu halten, wird neben der Qualität und Größe der Videodatei nun auch die Decodierenergie betrachtet. Diese Decodierenergie beschreibt den zus" Ã¤tzlichen Energieverbrauch den das mobile Endgerät zum Abspielen des Videos benötigt. Ein Modell, um diese Energie zu schätzen wurde in [HSR$^+$] vorgestellt und ist in Formel (1.1) zu sehen.

$$\hat{E}_{dec} = \sum_{\forall i} n_i \cdot e_i \tag{1.1}$$

Das Modell berechnet die Decodierenergie $\hat{E}_{dec}$ aus der Summe der einzelnen Energien die für die Features, beispielsweise der Prädiktion eines Blockes, benötigt wird. Der Index für ein Feature ist i. Für jedes Feature wird dann die Auftrittshäufigkeit n_i mit der spezifischen Energie e_i multipliziert, die vorher durch ein sogenanntes Training festgelegt wird. Diese spezifischen Energiewerte gilt es im Folgenden hinsichtlich der zuverlässigen Bestimmung zu untersuchen und verschiedene Kombinationen für die Bestimmung zu bewerten. Dafür wurden in Kapitel 2 sogenannte Konfidenz-Intervalle eingeführt. Die Grenzen dieser Intervalle legen fest, in welchem Energieintervall die spezifischen Energiewerte liegen sollten. Mit diesen Grenzen werden die Kombinationen miteinander verglichen und hinsichtlich

der zuverlässigen Bestimmung der Energiewerte bewertet. In Kapitel 3 wurden die 1%-Intervalle eingeführt, die ebenfalls zur Bewertung der Kombinationen herangezogen wurden. Die Grenzen dieser Intervalle sind jeweils 1% über dem Mittelwert oder unter dem Mittelwert der einzelnen Operationen. Der Unterschied zu den Konfidenz-Intervallen ist, dass die Grenzen lediglich vom Mittelwert abhängen und nicht von anderen Größen, wie beispielsweise der Menge der betrachteten Werten oder der Standardabweichung der Werte. Außerdem wurden in Kapitel 3 die verschiedenen Features hinsichtlich ihrer Zuverlässigkeit in der Bestimmung und ihrer Represäntativität in drei Bewertungsgruppen unterteilt. Im letzten Kapitel wurden neue Trainingssets erstellt und mit den 1%-Intervallen bewertet.

1.2 Trainings Setup

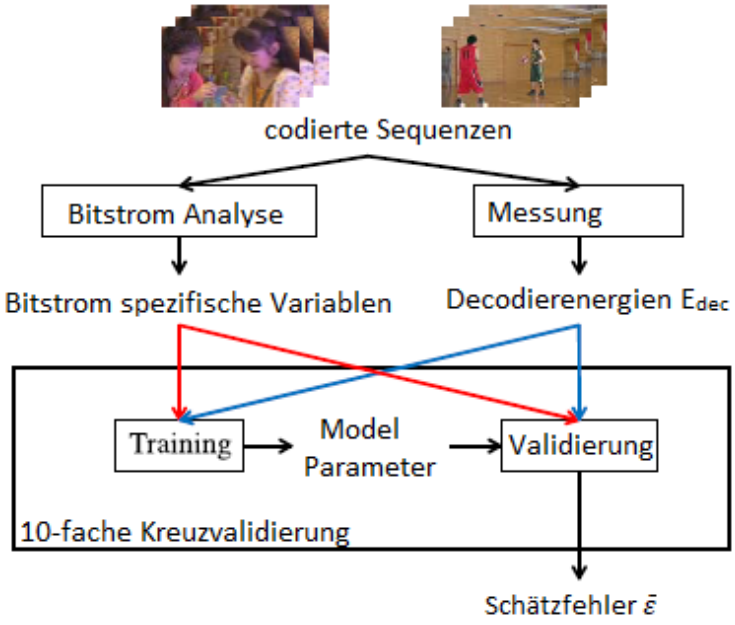

Abbildung 1.1: Trainingssetup zur Bestimmung der spezifischen Energiewerte [HSR+].

Das Trainingssetup für die Energiemodelle, welches zur Bestimmung der spezifischen Energien verwendet wird, ist in Abbildung 1.1 zu sehen. Für das Training werden verschiedene Sets von codierte Sequenzen verwendet, welche im Folgenden als Trainingssets bezeichnet werden. Durch diese Sequenzen können mittels Bitstromanalyse die Auftrittshäufigkeiten n_i bestimmt werden, die zum einen für das Training und zum anderen für die Validierung genutzt werden. Außerdem werden im Training und in der Validierung noch die gemessenen Decodierenergien der Sequenzen verwendet. Im Training werden die Parameter mit Hilfe von mathematischen Optimierungsverfahren bestmÃ¶glich angepasst. HierfÃ¼r wird die Methode der Kleinsten Quadrate angewandt. Diese wird durch die Matlab-Funktion

"lsqcurvefit"gemacht [Mat18a]. Nach dem Training wird, je nach gewähltem Kreuzvalidierungsfaktor k (in dieser Arbeit k= 1, 5, 10, 25), eine Kreuzvalidierung durchgeführt. Dabei wird die Menge aller Daten in k möglichst gleichgroße Teilmengen unterteilt. In k Iterationen wird immer eine Teilmenge herausgenommen. Aus den restlichen Teilmengen werden dann die spezifischen Variablen mittels Kurvenanpassung bestimmt und das Resultat dann mit der herausgenommenen Teilmenge der mittlere Schätzfehler bestimmt (Validierung). Aus allen Iterationen wird dann der Schätzfehler gemittelt, um den relativen Schätzfehler zu bekommen, welcher beschreibt, ob das Modell in der Praxis zu verwenden ist. In dieser Arbeit wurde auch eine Leave-One-Out Kreuzvalidierung verwendet (bei k=-1). Bei diesem Spezialfall wird in jeder Iteration aus der gesamten Menge der Daten nur die Daten einer Sequenz herausgenommen. Folglich ist die Anzahl der Iterationen gleich der Anzahl der Datensätze und das Trainingsset gleich dem Validierungsset.

1.3 Modelle

Für das testen von verschiedenen Modellen wurden die drei Modelle "BF27", "BFcomplete" und "BFHL" verwendet. Diese sind Modelle für die Schätzung der Decodierenergie gemäß der Formel (1.1) und werden daher als "feature based"(BF) bezeichnet. Sie unterscheiden sich in der Anzahl der Features. Das Modell "BF27" umfasst 27 Features, das Modell "BFcomplete" 180 und "BFHL" 31. Das Modell "BFcomplete" umfasst die meisten Features, da es Features besitzt die viele Coding-Modi und Blockgößen beschreiben. In "BF27" wurden einige Features aus dem Modell "BFcomplete" zusammengefasst und weniger Blockgrößberücksichtigt. Im Anhang in Tabelle 6.1 und 6.2 ist eine Auflistung der Modelle bzgl. ihrer Features zu finden. In [HSR$^+$] wurde gezeigt, dass dieses Modell jedoch einen geringeren Schätzfehler als das Modell "BFcomplete" aufweist. Das Modell "BFHL" ist ein sogenanntes hybrid Modell, welches eine Mischung aus dem Modell "BF27" und dem Modell "high-level linear model", welches in [HSR$^+$] beschrieben wurde, ist. Dieses Modell besitzt zusätzlich zu den Features aus "BF27" auch noch die Features Breite(der Frames), Höhe(der Frames), Anzahl der Frames und die Größe der Datei. Im folgenden werden neben den verschiedenen Modellen, Trainingssets und k-Werten verschiedene Softwaresetups verwendet. Wenn das Softwaresetup nicht explizit erwähnt wird, wurde das Softwaresetup "panda_FFMPEG_dualCore" verwendet.

1.4 Trainingssets

Die Trainingssets beschreiben eine Gruppe von Bitströmen. Diese werden durch das codieren von verschiedenen Sequenzen mit vier verschiedenen Quantisierungsparameter (QP=22,27,32,37) erhalten. Diese Bitströme werden dann für das, in Kapitel 1.2 beschriebene, Trainingssetup benötigte. Es werden

im folgenden verschiedene Traininssets verwendet, da jedes Trainingsset unterschiedlich konstruiert wurde und folglich verschiedene Eigenschaften besitzt.

1.4.1 Trainingssets realistic & realLarge & realintra

Für das "realistic" Trainingsset wurden die 32 Videosequenzen aus dem Anhang verwendet und wurden mit einem FFMPEG Encoder [Bel] mit den Encodereinstellungen aus [HSR+] generiert. Daher enthält das Set "realistic" $4 \cdot 32 = 128$ Bitströme. Beim Erstellen des Trainingssets realLarge wurde zusätzlich ein zweiter Encoder, HM Encoder [VC] mit randomacess- und intra-Einstellungen, verwendet und es umfasst 640 Bitströme. Beim Trainingsset "realintra"wurden alle 128 Bitströme mit einem HM Encoder mit Intra-Einstellungen codiert und es umfasst 128 Bitströme. [Her].

1.4.2 specific

Das Trainingsset ¨specific¨ wurde erstmals in [HSEK] vorgestellt. Für dieses Set wurden für jedes Feature ein spezieller Bitstrom erstellt, der das Verhalten hinsichtlich der Decodierenergie eines Features repräsentiert. Für das Trainingsset werden dann zwei Bitströme benötigt. Beispielsweise wird ein Bitstrom der mit einem herkömmlichen Encoder codiert wird und ein zusätzlicher Bitstrom für jedes Feature, bei dem alle Werte die größer bzw. kleiner 1 bzw -1 sind zu 1 bzw. -1 gesetzt werden, benötigt. Dann wird die Decodierenergie der beiden Bitströme gemessen und dann die Differenz vom ersten Bitstrom zum zweiten gebildet. Aus dem Quotienten dieser Differenz und der Summe der logarithmischen Differenzen der einzelnen Koeffizienten lässt sich dann die spezifische Energie berechnen. Dieses Set enthält 114 Bitströme, die mit hohem Zeitaufwand manuell erzeugt wurden. Allerdings ist es unbekannt, ob diese Bitströme repräsentative Werte für realistische Sequenzen, wie zum Beispiel im Set ¨realistic¨, widerspiegeln.

1.4.3 Trainingssets modified & mod27

Beim Trainingsset "modified¨ wird ein modifizierter Encoder, der verschiedene Features bevorzugt oder verweigert, benutzt, durch den die Bitströme zwei Vorteile bieten. Zum einen werden echte Sequenzen codiert und zum anderen werden je nach Codiermodus einzelnen Features bevorzugt. Daher ist dieses Set sozusagen eine Mischung zwischen dem Set ¨realistic¨ und dem Set ¨specific¨. Das Set ¨modified¨ umfasst dabei 712 Bitströme und das Set ¨mod27¨ umfasst lediglich 112 Bitströme, da dieses hinsichtlich der Features von BF27 erstellt wurde.

1.4.4 Trainingssets realSpec & altogether

In [Tei] wurden noch zwei weitere Sets "realspec" und "altogether" vorgestellt, welche verschiedene Sets miteinander vereint. Beim Set "realspec" wurden die Trainingssets "realistic" und "specific" miteinander vereint, so dass dieses Set 242 Bitströme beinhaltet. Das Set "altogether" vereint alle Trainingssets miteinander und umfasst 1494 Bitströme. Alle hier betrachteten Trainingssets sind in der Tabelle 1.1 zusammengefasst.

Trainingsset Index	Name	Anzahl der Bitströme
1	modified	712
2	realistic	128
3	realintra128	128
4	specific	114
5	realspec	242
6	realLarge	640
7	mod27	112
8	altogether	1494

Tabelle 1.1: Übersicht der verwendeten Trainingssets [Her].

Kapitel 2

Vergleich der Kombinationen

In diesem Kapitel werden die spezifischen Energiewerte e_i, die mittels verschiedener Trainingssets bestimmt werden, hinischtlich ihrer Zuverlässigkeit untersucht. Als erstes Kriterium für das Bewerten der spez. Energiewerte, wird das Konfidenz-Intervall aus [HSR$^+$] verwendet. Dabei beschreibt die Anzahl wie oft ein Feature außerhalb dieses Intervalls liegt, wie unzuverlässig die Energiewerte durch die Kombinationen aus Modell, Trainingsset und der Iterationen der Kreuzvalidierung bestimmt werden können. Daher sind Kombinationen, die wenige Werte außerhalb dieses Intervall aufweisen, zuverlässiger in der Bestimmung. Die Intervall-Grenzen, c_{left} die untere Grenze und c_{right} die obere Grenze, berechnen sich aus dem Mittelwert $\bar{x}$ der Werte, der Standardabweichung σ, der Anzahl der Werte m und einer Wahrscheinlichkeit α, welche die Wahrscheinlichkeit angibt, dass μ im Intervall zwischen c_{left} und c_{right} liegt (hier 95%). Die Funktion t_α entspricht der Student-t-Verteilung [Mat18b] tinv", die eine inverse kumulative Verteilungsfunktion ist.

$$c_{left} = \bar{x} - \frac{\sigma}{\sqrt{m}} \cdot t_a(m-1) < \mu < c_{right} = \bar{x} + \frac{\sigma}{\sqrt{m}} \cdot t_a(m-1) \tag{2.1}$$

2.1 Konfident-Grenzen-Check mit Trainingssets

Im folgenden Abschnitt werden lediglich die die acht Werte (m=8) aus den verschiedenen Trainingssets verwendet, um den Mittelwert, die Standardabweichung und das Konfidenzintervall zu berechnen. Nach dem Berechnen des Konfidenzintervalls werden die Werte der einzelnen Trainingssets und Features geprüft, ob sie im, für dieses Feature berechnete, Konfidenzintervall liegen. Zwei Beispiele sind in Abbildung 2.1 zu sehen. Zum einen, links, das Feature "E0" mit dem Modell BF27 und einem k=10 und zum Anderen, rechts, das Feature "PBSlice" mit dem Modell BFcomplete und einem k=10. Die untere Konfidenz-Grenze ist in den Grafiken in schwarz dargestellt, die Obere in rot.

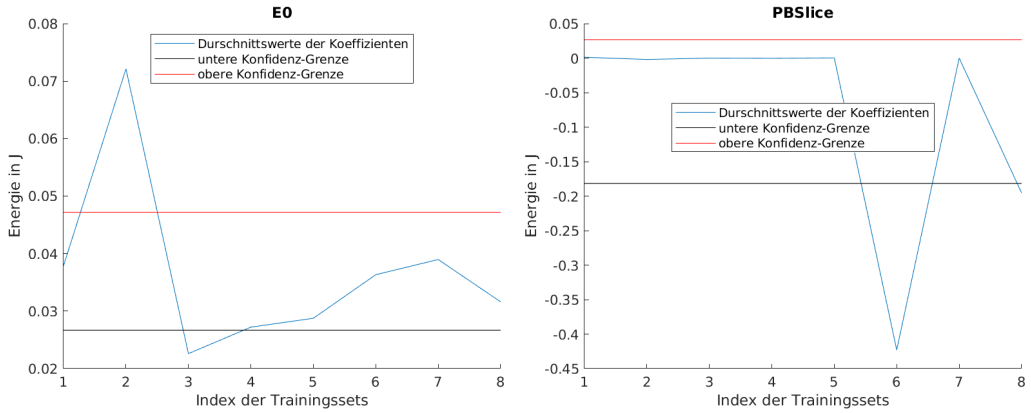

Abbildung 2.1: Durchschnittswerte der Features E0 (links, Modell BF27) und PBSlice (rechts, Modell BFcomplete) der Trainingssets bei k=10 und die dazugehörigen Konfidenz-Grenzen

Die blauen Kurven beschreiben die gemittelten Werte des Features bei der Kombination des dem Index (siehe Tabelle 1.1) entsprechenden Trainingssets. Beispielsweise liegt der Wert vom Feature E0 (in der linken Grafik zu sehen), der durch das Trainingsset "modified" bestimmt wurde bei etwa 0.039 Joule und ist somit innerhalb der Grenzen. Hier ist der Wert von E0 zuverlässig durch die Kombination vom Modell "BF27", dem Trainingsset "modified" und einem k=10 bestimmt worden, mit dem Trainingsset "realistic" hingegen nicht. In der rechten Grafik ist zu sehen, dass lediglich die Trainingssets "realLarge" und "altogether" den spezifischen Energiewert für das Feature "PBSlice", in Kombination mit dem Modell "BFcomplete" und k=10, nicht zuverlässig bestimmen. Außerdem ist zu sehen, dass die Energie, die für das Feature "PBSlice" aufgwendet werden muss, negativ ist und somit ein Energiegewinn vorliegen würde, was beim Ausführen von Operationen in der Praxis nicht passieren kann. Dieser negative Wert kann durch ein Overfitting Problem im Training zustande kommen.

Die folgenden Tabellen zeigen wie viele Features in einem Trainingsset zu klein (n_L) oder zu groß n_H sind. Außerdem sieht man die komplette Anzahl der Features eines Trainingssets, die nicht im Konfidenzintervall liegen (n_O) und die Prozentzahl, die angibt wie viel Prozent der Features in diesem Trainingsset nicht im Intervall lagen (n_O/n). n entspricht der Gesamtanzahl der Features.

2.1.1 Model: BF27 (27 Features)

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	3	10	6	1	10	4	0	10
n_H	10	9	6	9	9	15	7	9
n_O	13	19	12	10	19	19	7	19
n_O/n	48.1%	70.4%	44.4%	37.0 %	70.4%	70.4%	25.9%	70.4%

Tabelle 2.1: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BF27"und k=1).

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	3	12	2	2	0	3	0	3
n_H	7	13	6	5	8	13	7	10
n_O	10	25	8	7	8	16	7	13
n_O/n	37.0%	92.6%	29.6%	25.9%	29.6%	59.3%	25.9%	48.1%

Tabelle 2.2: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BF27"und k=10).

Wie man aus Tabelle 2.1 und Tabelle 2.2 entnehmen kann sind die Ergebnisse von den Trainings-Sets: "mod27", und "realspec"insgesamt am zuverlässigsten, dies ist an der Rate n_O/n zu sehe, da diese bei beiden Trainingssets am geringsten ist.

Das Trainings-Set "realistic"und "realLarge"sind mit 19 Werten (70.4%) außerhalb der Grenzen am unzuverlässigsten in der Bestimmung der Energiewerte. Außerdem ist zu erkennen, dass die Werte in Tabelle 2.2 allgemein niedriger sind. Dies ist darauf zurückzuführen, dass es zehnmal so viele Werte gibt, wodurch die gemittelte Fehlerrate sich verringert. Dies bedeutet, dass je mehr Iterationen der Kreuzvalidierung durchgeführt werden, desto mehr Werte liegen prozentual im Intervall. Als nächstes werden die selben Kombinationen von Modell und Anzahl der Iterationen der Kreuzvalidierung betrachtet, allerdings wird statt dem Softwarsetup "panda_FFMPEG_dualCore"das Setup "panda_HM" verwendet.

Training: panda_HM

Set	modified	realistic	realintra128	specific	realspec	realLarge
n_L	1	13	12	3	13	2
n_H	8	5	3	7	5	7
n_O	9	18	15	10	18	9
n_O/n	33.3%	66.7%	55.6%	37%	66.7%	33.3%

Tabelle 2.3: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BF27"und k=1).

Set	modified	realistic	realintra128	specific	realspec	realLarge
n_L	1	10	11	2	0	1
n_H	3	8	2	2	2	4
n_O	4	18	13	4	2	5
n_O/n	14.8%	66.7%	48.1%	14.8%	7.4%	18.5%

Tabelle 2.4: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BF27"und k=10).

Es ist in Tabelle 2.3 und 2.4 zu sehen, dass mit dem Softwaresetup "panda_HM"sich die Energiewerte zuverlässiger bestimmen lassen als mit dem Setup "panda_FFMPEG_dualCore"(Tabelle 2.1 und 2.2) erzielt werden. Im Folgenden wird wieder das Softwaresetup "panda_FFMPEG_dualCore"und statt dem Modell "BF27" das Modell "BFcomplete", welches mehr Features aufweist, verwendet.

2.1.2 Model: BFcomplete (184 Features)

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	0	8	1	1	8	71	6	8
n_H	2	6	2	3	6	94	7	6
n_O	2	14	3	4	14	165	13	14
n_O/n	1.09%	7.61%	1.63%	2.17%	7.61%	89.7%	7.07%	7.61%

Tabelle 2.5: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BFcomplete"und k=1).

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	0	8	1	3	1	91	7	88
n_H	0	10	0	6	2	67	7	63
n_O	0	18	1	9	3	158	14	151
n_O/n	0.0%	9.78%	0.543%	4.89%	1.63%	85.9%	7.61%	82.1%

Tabelle 2.6: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BFcomplete"und k=10).

In Tabelle 2.5 ist zu erkennen, dass das Trainings-Set "realLarge"mit dem Model BFcomplete ungeeignet ist, da 89.7% aller Werte nicht in den berechneten Grenzen liegen. Die anderen Sets hingegen erzielen alle akzeptable Ergebnisse mit maximal 14 Werten die außerhalb der Grenzen liegen, das sind etwa 8.7% aller dieser Werte.

Bei einem k = 10 (Tabelle 2.6) verschlechtert sich das Ergebnis bei "altogether"enorm (von 4.3% auf 75%), außerdem verschlechtert sich auch das Set "realistic"um das doppelte. Bei den anderen Sets kann man jedoch Verbesserungen oder gleiche Ergebnisse feststellen. Vor allem beim Set "modified ¨ ist kein Wert von 184 außerhalb des Konfidenzintervalls. Als letztes Modell ist nun das Modell "BFHL" hinsichtlich der Zuverlässigkeit in der Bestimmung mit verschiedenen Trainingssets zu sehen.

2.1.3 Model: BFHL (31 Features)

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	2	9	3	1	9	10	2	9
n_H	5	6	12	9	6	6	4	6
n_O	7	15	15	10	15	16	6	15
n_O/n	22.6%	48.4%	48.4%	32.3%	48.4%	51.6%	19.4%	48.4%

Tabelle 2.7: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BFHL"und k=1).

Set	modified	realistic	realintra128	specific	realspec	realLarge	mod27	altogether
n_L	11	6	4	0	0	3	11	1
n_H	3	18	8	6	6	6	5	5
n_O	14	24	12	6	6	9	16	6
n_O/n	45.2%	77.4%	38.7%	19.4%	19.4%	29.0%	51.6%	19.4%

Tabelle 2.8: Häufigkeit dass Werte außerhalb der berechneten Grenzen liegen (mit dem Model "BFcomplete"und k=10).

In Tabelle 2.7 wird festgestellt, dass das Model "BFHL"bei einem k=1 keine guten Ergebnisse erzielt (min 19.4% der Werte liegen außerhalb der Grenzen). Bei einem k=10 ist zu sehen, dass sich die Energiewerte mit den Trainingssets "realintra128", "specific", "realspec ", "realLarge "und "altogether"zuverlässiger bestimmen lassen. Bei den anderen Trainingssets jedoch verschlechtert sich die Bestimmung der Werte.

Die oben genannten Ergebnisse geben erstmals einen Eindruck, welche Trainingssets mit welchen Modellen besser zusammenpassen. Allerdings sind diese Werte nicht sehr aussagekräftig, da die Grenzintervalle im Vergleich zum Mittelwert der Features ziemlich groß sind. In [HSR$^+$] wurde daher eine Bedingung aufgestellt,

$$\Delta c = c_{right} - c_{left} = 2 \cdot \frac{\sigma}{\sqrt{m}} \cdot t_a(m-1) < \beta \cdot \bar{x} \tag{2.2}$$

in der β einen kleinen Wert (im Folgenden 0,01) besitzt, um aussagekräftige Ergebnisse zu erzielen. Um ein Konfidenz-Intervall zu bekommen, das die oben genannte Bedingung erfüllt, ist es hilfreich, wenn die Anzahl der Messwerte m groß ist [HSR$^+$]. Entsprechende Analysen werden im nächsten Unterkapitel vorgestellt.

2.2 Konfident-Grenzen-Check mit verschiedener Anzahl der Iterationen der Kreuzvalidierung

Um ein größeres m und aussagekräftigere Konfidenzintervalle zu bekommen, wurden statt den acht Mittelwerten der jeweiligen Trainingssets, die einzelnen Koeffizienten aus den Iterationen der Kreuzvalidierung genommen und für jedes Trainingsset wurde separat ein Konfidenzintervall pro Feature berechnet. Die Formel ist die selbe, die bereits oben verwendet wurde (Formel (2.1)), allerdings variiert das m je nach dem welcher Wert für k gewählt wird. Beispielsweise werden bei einem k=25 25 Werte für das Feature PBSlice erhalten (Trainingsset: altogether, Model:BF27). Aus diesen 25 Werten wird dann der Mittelwert $\bar{x}$ und die Standardabweichung σ berechnet und für m der Wert 25, da 25 Werte des Features vorhanden sind, genommen. Aus den nun gegebenen Größen können dann die Konfidenz-Grenzen für das Feature berechnet werden. Das Beispiel ist in Abb. 2.3 a) und ein Weiteres in Abb. 2.3 b) zu sehen.

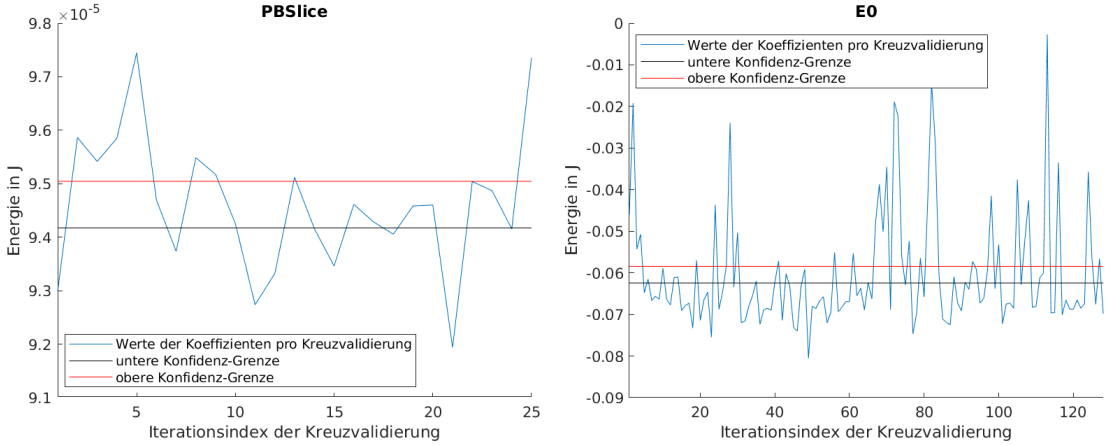

Abbildung 2.2: Einzelne Werte der Koeffizienten des Features PBSlice (bei k=25, Trainingsset=altogether, Model=BF27, Softwaresetup=panda_FFMPEG_DualCore) und die zugehörigen Konfidenzgrenzen in a) und einzelne Werte der Koeffizienten des Features E0 (bei k=-1, Trainingsset=realistic, Model=BFcomplete, Softwaresetup=panda_FFMPEG_DualCore) und die zugehörigen Konfidenzgrenzen in b)

Im Folgenden wurden wieder für k die Werte 1, 5, 10, 25 und -1 verwendet. So wurden für die Trainingssets ein, fünf, zehn oder 25 Werte pro Feature errechnet. Beim Sonderfall mit k = -1 (siehe Abschnitt 1.2) erhält man deutlich mehr Werte pro Feature. Beispielsweise in Abbildung 2.2 auf der rechten Seite ist zu sehen, dass es 128 Werte für diese Kombination für das Feature E0 gibt. Aus diesen Werten (ausgenommen die Werte mit k=1, da diese nur einen Wert haben und sich somit kein

Modell	BF27						
k	5		10		25		
Trainingsset	realspec	altogether	realspec	altogether	realspec	altogether	modified
n_O/n	35.6%	36.3%	54.8%	54.1%	66.4%	71.1%	56%

Modell	BF27						
k	25		-1				
Trainingsset	specific	realintra128	realspec	altogether	modified	specific	realintra128
n_O/n	42.2%	36.6%	49.8%	78.5%	62.5%	37.5%	39.1%

Modell	BF27	BFHL					
k	-1	5	10		25		
Trainingsset	mod27	altogether	realspec	altogether	realspec	altogether	specific
n_O/n	56.9%	34.2%	55.5%	53.9%	72.7%	69.1%	43.9%

Modell	BFHL		
k	-1		
Trainingsset	realspec	altogether	specific
n_O/n	57.5%	68.4%	37.5%

Tabelle 2.9: Kombinationen mit geeigneten Konfidenzintervallen

Konfidenzintervall berechnen lässt, da hier keine Kreuzvalidierung durchgeführt wird und das Trainingsset gleich dem Validierungsset ist) wurden dann die Konfidenzintervalle der einzelnen Features in jedem Trainingsset und jedes Features berechnet. Anschließend wurde geprüft, wie viele Konfidenzintervalle der oben eingeführten Bedingung entsprechen (Formel 2.2). Als Auswahlkriterium für aussagekräftige Ergebnisse wurde festgelegt, dass mindestens 75% aller Konfidenzintervalle der Features in einem Trainingsset (und einem Wert k) die Bedingung, aus (2.2), erfüllen müssen. Aus diesen Schritten ergeben sich dann 24 Kombinationen von Modell, Trainingsset und Anzahl der Iterationen k. In der nachfolgenden Tabelle sind die Kombinationen und die Prozentzahl der außerhalb der Grenzen liegenden Werten n_O/n (einzelne Werte aus der Kreuzvalidierung, die nicht im berechnetem Intervall des zugehörigen Features liegen) zu sehen.

In Tabelle 2.9 sind die verschidenen Kombinationen zu sehen, welche die Bedingung aus der Formel (2.2) erfÃ¼llen. Das heißt das β ist bei diesen kleiner als 1%. Zu sehen sind außerdem die Fehlerrate n_O/n der Kombinationen. Die zuverlässigsten Bestimmungen werden durch Kombinationen gemacht, die hier eine Fehlerrate von etwa 35% besitzen. Dieser Wert ist allerdings sehr groß, was bedeutet, dass lediglich etwa zwei Drittel aller spezifischen Energiewerte zuverlässig bestimmt werden können.

Kapitel 3

Grenzen-Check anhand der Mittelwerte

Da die Konfidenz-Grenzen stark von der Anzahl der verschiedenen Koeffizienten m und daher auch von der Anzahl der Kreuzvalidierungen k abhängen, wurden neue Grenzen berechnet. Um die Werte aus dem Trainings mit unterschiedlichem k-Wert besser zu vergleichen wurden die Grenzen wie folgt berechnet:

$$c_{left} = \bar{x} \cdot (1 - \gamma) < \mu < c_{right} = \bar{x} \cdot (1 + \gamma) \tag{3.1}$$

Dabei ist $\bar{x}$ wieder der Mittelwert des Features und γ ist die gewünschte prozentuale Genauigkeit von der die Werte maximal abweichen dürfen. In diesem Forschungspraktikum wurde für γ der Wert 0,01 (1%) gewählt. In der Tabelle 3.1 sind die Prozentzahlen eines jeden Modells zu sehen, welche beschreiben wie viele Koeffizienten von allen Features außerhalb der Grenzen lagen. Der Wert k=1 ist wieder nicht berücksichtigt, da es nur einen Koeffizienten gibt und dieser automatisch den Mittelwert bildet. In Abbildung 3.1 ist wieder ein Beispiel für das Feature PBSlice bei einem k von 25 (Trainingsset: ¨altogether¨, Modell: BF27) zu sehen. Die Energiewerte der einzelnen Koeffizienten in jedem Durchgang der Kreuzvalidierung sind durch die blaue Linie veranschaulicht und die berechneten 1% Grenzen sind in rot und schwarz zu sehen.

In Tabelle 3.1 ist zu sehen, dass vor Allem das Trainingsset ¨altogether¨ mit den Modellen BF27 und BFHL gute Ergebnisse erzielt. Grund dafür könnte sein, dass das Trainingsset deutlich am meisten Bitströme beinhaltet und somit das Fehlen einzelner Bitströme nicht so stark ins Gewicht fällt. Außerdem ist zu erkennen, dass eine Kreuzvalidierung mit k=-1 bessere Werte erzielt als mit anderen

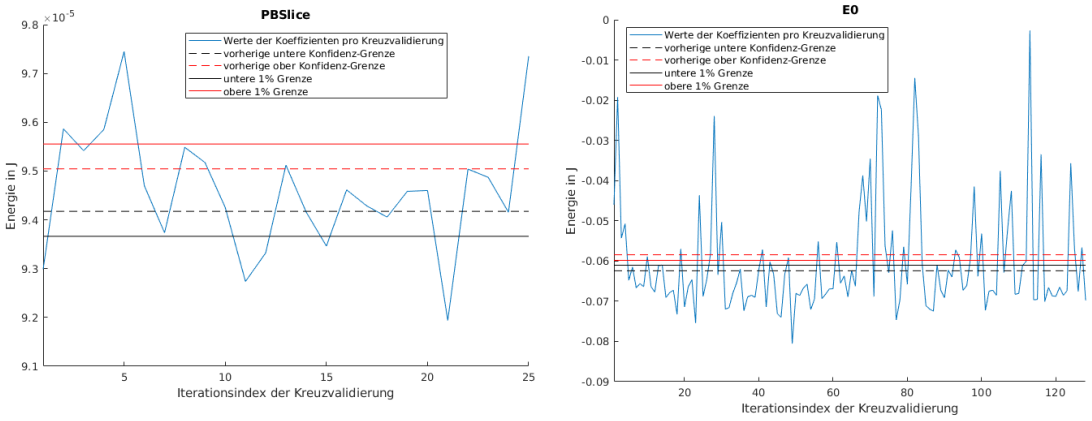

Abbildung 3.1: Einzelne Werte der Koeffizienten des Features PBSlice (bei k=25, Trainingsset=altogether, Model=BF27) und die zugehörigen 1%-Grenzen in a) und Einzelne Werte der Koeffizienten des Features E0 (bei k=-1, Trainingsset=realistic, Model=BFcomplete) und die zugehörigen 1%-Grenzen in b)

k. Auch hier könnte die Begründung sein, dass deutlich mehr Werte berücksichtigt werden und einzelne Bitströme nicht so starken Einfluss auf das Ergebnis haben. Mit diesem k wurden außerdem auch gute Ergebnisse mit den Trainingssets realspec und realintra128 erzielt, so wie mit dem Modelle BFcomplete in Kombination mit dem Trainingsset ¨altogether¨.

k	Trainingsset	BF27	BFcomplete	BFHL	k	Trainingsset	BF27	BFcomplete	BFHL
5	modified	88.9%	92%	93.5%	25	modified	73.3%	84.4%	80%
	realistic	97.8%	98.9%	96.1%		realistic	92.4%	95.5%	91.1%
	realintra128	58.5%	68%	61.9%		realintra128	42.8%	64.8%	56.8%
	specific	94.1%	97.6%	93.5%		specific	60.0%	98.1%	64.9%
	realspec	64.4%	90.9%	75.5%		realspec	32.1%	86.5%	54.1%
	realLarge	91.9%	94.8%	93.5%		realLarge	81.4%	94.1%	87.5%
	mod27	97.8%	96.6%	98.1%		mod27	87.9%	98.3%	88.1%
	altogether	17.0%	90.1%	20.6%		altogether	7.7%	89.8%	11.4%
10	modified	84.8%	89.9%	92.6%	-1	modified	7.4%	34.1%	49.7%
	realistic	95.6%	97.4%	92.6%		realistic	72.5%	92.4%	77.5%
	realintra128	54.4%	66.1%	61.9%		realintra128	23.3%	57.2%	45.1%
	specific	87.4%	97.9%	91%		specific	25.3%	88.7%	29.1%
	realspec	56.7%	91.1%	70%		realspec	16.4%	82.3%	24.4%
	realLarge	88.5%	93.8%	90.6%		realLarge	24.0%	90%	27.8%
	mod27	96.7%	95.3%	94.8%		mod27	62.4%	97.7%	68.8%
	altogether	14.4%	90.4%	19.7%		altogether	4.3%	8.2%	6.99%

Tabelle 3.1: Fehlerrate der verschiedenen Kombinationen, die aussagt wie viele der Koeffizienten nicht in den Grenzen liegen (mit Grün hinterlegt sind die Fehlerraten $< 25\%$)

Kapitel 4

Grenzen-Check von einzelnen Features

Bisher wurden lediglich die verschiedenen Trainingssets-Modell-k-Kombination hinsichtlich der Fehlerrate (n_O/n), wie viele Energiewerte prozentual außerhalb des Intervalles liegen, verglichen. Beispielsweise sind von allen durch das Modell BF27, dem Trainingsset "altogether" und fünf Iterationen der Kreuzvalidierung bestimmten Energiewerte 17% außerhalb des 1%-Intervalles (siehe Tabelle 3.1). Doch im folgenden werden die einzelnen Features hinsichtlich der Zuverlässigkeit bei ihre Bestimmung verglichen. Dazu wurden zu jedem Feature in jedem Modell eine Tabelle angelegt, in der die Fehlerrate, vom Grenzen-Check mit den 1%-Grenzen, für jede k-Trainingsset-Kombination eingetragen wurde (somit $4 \cdot 8 = 32$ Werte, 4 Werte für k und 8 verschiedene Trainingssets). Daraufhin wurde für jedes Feature der kleinste Wert, der größte Wert und der Mittelwert von allen 32 Werten herausgenommen bzw. berechnet. Der kleinste Wert, wurde dafür hergenommen die Zuverlässigkeit der Features in drei Gruppen zu unterteilen: "Super", "OK" und "Bad". Ein "Super" bekommt ein Feature wenn es mit mindestens einer k-Trainingsset-Kombination eine Fehlerrate von 0% aufweisen kann, also stets innerhalb des 1%-Intervalles liegt. Ein "OK" bekommt es, wenn die kleinste Fehlerrate über 0 und kleiner 10% liegt, und ein "Bad", wenn es über 10% liegt. In Abbildung 4.1 ist als Beispiel der Matlab-Table zu sehen in dem die Werte der Features stehen. In der ersten Spalte ist der Name des Features, in der zweiten Spalte die Tabelle mit den 32 Werten der Kombination und die Mittelwerte pro k, Trainingsset oder von allen. In Abbildung 3.2 rechts oben, ist die Tabelle mit den 32 Werten der Kombinationen zu sehen. Die ersten vier Zeilen beschreiben die Werte von k (5, 10, 25, -1) und die ersten acht Spalten stehen für die verschiedenen Trainingssets. Die Spalte neun ist der Mittelwert aus einer Zeile und die Zeile 5 der Mittelwert der Spalten, somit erhält man in Zeile 5 und Spalte 9 den gesamten Mittelwert der Fehlerrate für das jeweilige Feature. In der dritten Spalte ist die kleinste Fehlerrate des Features

zu finden, in der Vierten die Größte. Diese ist meistens 1 bzw. 100%, da es Kombinationen gibt, die die Features nicht wirklich bestimmen können. In Spalte Fünf ist der Mittelwert aller Fehlerraten des Features, in der Sechsten eine der drei, oben genannten, Bewertungen und in der Siebten ein Cell-Array, in dem die besten k-Trainingsset-Kombinationen zum Bestimmen dieses Feature stehen (Beispiel in Abbildung 3.2,rechts unten).

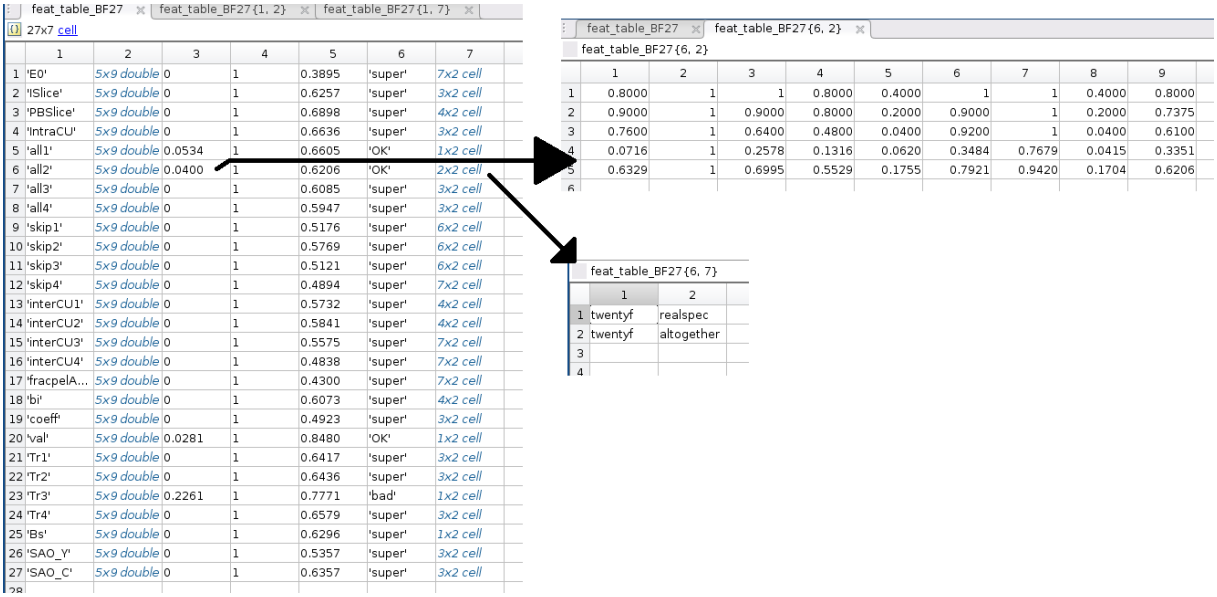

Abbildung 4.1: Übersicht der Features des Modelles BF27 mit den Bewertungskriterien für die Zuverlässigkeit zur Bestimmung

Aus der Abbildung 4.1 ist zu sehen, dass sich die meisten Features aus dem Model BF27 zuverlässig bestimmen lassen, da ihre minimale Fehlerrate gleich Null ist. Das heißt es gibt eine Kombination aus Trainingsset und k-Wert, um diesen Wert sicher in den Grenzen zu bestimmen. Die Features 'all1', 'all2' und 'val' hingegen lassen sich nicht sicher mit einer Kombination innerhalb der Grenzen bestimmen. Bei diesen Features steht im Cell-Array in Spalte Sieben die Kombinationen, mit denen das beste Ergebnis erzielt wird. Zuletzt gibt es noch ein Feature, dass sich deutlich weniger zuverlässig bestimmen lässt als die Anderen. Dieses Feature 'Tr3' besitzt eine Fehlerrate von 22.61% bei der Kombination, mit der es am besten bestimmt wurde. Die Werte dieses Features besitzen also eine große Streuung, so dass nur schlecht ein spezifischer Energiewert festgelegt werden kann.

Kapitel 5

Neue Trainingssets

Nachdem lediglich die acht Trainingssets aus Tabelle 1.1 hinsichtlich der Zuverlässigkeit zur Bestimmung der Features betrachtet worden sind, werden in diesem Kapitel neu erstellte Trainingssets vorgestellt. Ziel ist es ein neues Trainingsset zu erstellen, welches wie beispielsweise das Trainingsset "altogether" gute Ergebnisse erzielt, jedoch weniger Bitströme verwendet und somit den Aufwand bzw. Zeit bei der Berechnung spart.

Trainingsset	Anzahl der Bitströme
cut_all	1092
cut_all_inverse	408
nonzero27	578
realspecmod27	354
realspecmodified	954

Tabelle 5.1: Übersicht der neu erstellten Trainingssets.

Bei den ersten zwei Trainingssets "cut_all" und "cut_all_inverse" wurde als Grundlage das Trainingsset "altogether" verwendet. Für das Set "cut_all" wurden zufällig mehrere Bitströme aus "altogether" herausgenommen, so dass dieses Set 1092 Bitströme enthält. Alle herausgenommenen Bitströme sind im Set "cuttet_all_inverse" zu finden, welches 408 Bitströme enthält. Würden beide Sets wieder zusammengeführt werden würden sie wieder das Set "altogether" ergeben.
Das nächste Set ist das Set "nonzero27". Bei diesem Set wurden alle Bitströme aus "altogether" verwendet, bei denen die Auftrittshäufigkeiten n_i für alle Features aus dem Modell BF27 ungleich Null sind. Somit enthält das Set "nonzero27" 578 Bitströme.
Zuletzt wurden noch die Sets "realspecmod27" (354 Bitströme) und "realspecmodified" (954 Bitströme) erstellt, welche die Zusammenführungen von "realspec" mit den Sets "mod27" bzw. "modified" sind. Der Gedanke dahinter war, dass diese Sets, wie auch "altogether", Bitströme enthalten die, wie in Kapitel 1.4 beschrieben, unterschiedlich erstellt wurden und unterschiedliche Eigenschaften

besitzen.

alle neu erstellten Trainingssets sind in Tabelle 4.1 zu sehen. Um herauszufinden, ob die neuen Trainingssets gute Ergebnisse erzielen, wurde wieder der Grenzen-Check mit den 1%-Grenzen verwendet (vgl. Kapitel 3). Die Ergebnisse sind in Tabelle 4.2 zu sehen. In Dieser Tabelle sind wieder die Fehlerraten der einzelnen Kombinationen von k, Modell und Trainingsset verzeichnet.

k	Trainingsset	BF27	BFcomplete	BFHL
5	cut_all	88.1%	93%	92.9%
	cut_all_inverse	93.3%	94.9%	94.2%
	nonzero27	91.1%	94.7%	94.8%
	realspecmod27	89.6%	94.3%	87.1%
	realspecmodified	85.2%	91.1%	85.8%
10	cut_all	84.8%	94.7%	88.1%
	cut_all_inverse	85.2%	94.4%	89%
	nonzero27	84.4%	94.8%	94.5%
	realspecmod27	94.3%	93.5%	84.5%
	realspecmodified	79.6%	89.7%	79.4%
25	cut_all	72%	94%	76.1%
	cut_all_inverse	74.7%	93.8%	78.6%
	nonzero27	77.8%	94.6%	90%
	realspecmod27	67.1%	94.5%	68.9%
	realspecmodified	63.9%	92.7%	64.5%
-1	cut_all	10.6%	88.7%	14.3%
	cut_all_inverse	14.1%	90.7%	17.8%
	nonzero27	18.9%	90.2%	55.7%
	realspecmod27	13.3%	90.8%	15.6%
	realspecmodified	5.2%	87.9%	6.1%

Tabelle 5.2: Übersicht der Fehlerraten der neu erstellten Trainingssets mit verschiedenen k und verschiedenen Modellen (mit Grün hinterlegt sind die Fehlerraten $< 25\%$).

In Tabelle 4.2 ist zu sehen, dass alle neuen Sets gute Ergebnisse mit dem Modell BF27 und einem k von -1 erzielen. Mit den anderen Werten von k sind die Ergebnisse deutlich schlechter als die Ergebnisse von "altogether". Mit dem Modell BFcomplete wurden, wie auch in Tabelle 3.1 die schlechtesten Ergebnisse erzielt. Beim Modell BFHL wurde bei einem k=-1 ein besseres Ergebnis mit dem neuen Set "realspecmodified" erzielt, allerdings wurden bei anderen Werten für die Kreuzvalidierung wieder schlechtere Ergebnisse erzielt. Am besten für eine zuverlässige Bestimmung der spezifischen Energiewerte sind daher Trainingssets die möglichst viele Bitströme beinhalten und Bitströme die unterschiedlich generiert wurden. Beispiele dafür sind "altogether" und "realspecmodified". Bei den Modellen sind die Modelle BF27 und BFHL zu empfehlen. BFcomplete ist in der Bestimmung nicht sehr zuverlässig, dies könnte an der großen Anzahl an Features liegen, von denen mehrere schwer zu bestimmen sind.

Kapitel 6

Zusammenfassung

In dieser Arbeit wurden verschiedene Kombinationen aus Modellen, Trainingssets und Anzahl der Iterationen der Kreuzvalidierung hinsichtlich der zuverlässigen Bestimmung der spezifischen Energiewerte der Features untersucht. Zuerst wurde das Trainingssetup, die Modelle und die verschiedenen Trainingssets vorgestellt. Danach wurden ein Konfidenz-Intervall vorgestellt, welches herangezogen wurde, um die verschiedenen Kombinationen zu bewerten. Dabei wurden zuerst die Mittelwerte der Energiewerte die bei den verschiedenen Trainingsset bestimmt wurden betrachtet. Allerdings waren die Konfidenz-Intervalle zu groß, so dass diese nicht repräsentativ für die Zuverlässigkeit der Bestimmung waren. Folglich wurden für die Berechnung der Konfidenzintervalle nicht die Mittelwerte der Kombinationen verwendet, sondern die exakten Koeffizienten eine jeden Iteration der Kreuzvalidierung. Dadurch wurden die Intervalle repräsentativer. Neben den Konfidenz-Intervall wurde ein weiteres Intervall zur Bewertung der Kombinationen eingeführt, welches Grenzen, die abhängig vom Mittelwert der Koeffizienten der Features sind, verwendet. Dieses Intervall wurde für die Bewertung hinsichtlich einer zuverlässigen Bestimmung als geeigneter betrachtet. Des weiteren wurden die einzelnen Features der Modelle betrachtet. Dabei wurden die Features in drei Kategorien unterteilt, welche die zuverlässige Bestimmung der einzelnen Features bewertet. Zuletzt wurden aus den Erkenntnissen dieser Arbeit weiter Trainingssets generiert, welche zur zuverlässigen Bestimmung verwendet werden sollten.

Literaturverzeichnis

[Bel] Bellard, F.: FFmpeg. In: *http://ffmpeg.org/* , aufgerufen Juni 2017

[Bos] Bossen, F: Common HM test conditions and software reference configurations. In: *document JCTVC-L1100* , Genf, Schweiz, Januar 2013

[Her] Herglotz, Christian: Energy Efficient Video Decoding. In: *Dissertation, Friedrich-Alexander-Universität (FAU), Verlag Dr.Hut* , 2018

[HSEK] Herglotz, C. ; Springer, D. ; Eichenseer, A. ; Kaup, A.: Modeling the energy consumption of HEVC intra decoding. In: *20th International Conference on Systems, Signals and Image Processing (IWSSIP)* , Bukarest, Rumänien, July 2013, S. 91–94

[HSR+] Herglotz, C. ; Springer, D. ; Reichenbach, M. ; Stabernack, B. ; Kaup, A.: Modeling the Energy Consumption of the HEVC Decoding Process. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28, Januar 2018, Nr. 1, S. 217–229

[HWD+] Herglotz, C. ; Wen, Y. ; Dai, B. ; Kränzler, M. ; Kaup, A.: Modeling the Energy Consumption of the HEVC Decoding Process. In: *Picture Coding Symposium (PCS)* , Nürnberg, Germany, Dezember 2016

[Mat18a] MathWorks: lsqcurvefit: Solve nonlinear curve-fitting (data-fitting) problems in least-squares sense. In: *https://de.mathworks.com/help/optim/ug/lsqcurvefit.html?s_tid=doc_ta* (aufgerufen am 10.08.2018)

[Mat18b] MathWorks: tinv: Student's t inverse cumulative distribution function. In: *https://de.mathworks.com/help/stats/tinv.html* (aufgerufen am 10.08.2018)

[MTAF] Mallikarachchi, T. ; Talagala, D.S. ; Arachchi, H.K. ; Fernando, A.: Decoder energy-aware intra-coded HEVC bit stream gerneration. In: *IEEE International Conference on Multimedia and Expo (ICME)* , Seattle, USA, Juli 2016, S. 1–6

[SA] Schaar, M. van d. ; Andreopoulos, Y.: Rate-distortion-complexity modeling for network and receiver aware adaption. In: *IEEE Transactions on Multimedia and Expo (ICME)* , Juni 2005, Nr. 7(3), S. 471–479

[Sta17] Statista: Umfrage Anzahl der Smartphonenutzer in Deutschland seit 2010. In: *https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010* (aufgerufen am 09.August 2017)

[Tei] Teiluf, Valentina: Untersuchung der Decodierenergie-Raten-Verzerrungs-Optimierung für Videosequenzen (Investigation of Decoding-Energy-Rate-Distortion Optimization for Video Sequences). In: *Bachelor's Thesis, Multimedia Communications and Signal Processing, Friedrich-Alexander-University Erlangen-Nürnberg (FAU)* , September 2016

[uni] university, Arizona state: YUV Video Sequences. In: *http://trace.eas.asu.edu/yuv/* , 2010

[VC] Video Coding, Joint Collaborative T.: HEVC test model reference software (HM). In: *http://hevc.hhi.frauenhofer.de/* , aufgerufen Juni 2017

Anhang

Index	Referenz	BF27	BFcomplete	BFHL
E0	[HSR+]	1	1	1
N	[HWD+]	-	1	-
ISlice	[HWD+]	1	1	1
PBSlice	[HWD+]	1	1	1
Iblocks	[HSR+]	1	1	1
IPblocks	[HSR+]	4	4	4
ILpla	[HSR+]	-	4	-
ILdc	[HSR+]	-	4	-
Ihvd	[HSR+]	-	4	-
ILang	[HSR+]	-	4	-
ILhor	[MTAF]	-	4	-
ILver	[MTAF]	-	4	-
ILang2	[MTAF]	-	4	-
ILang18	[MTAF]	-	4	-
ILang34	[MTAF]	-	4	-
ILremhor	[MTAF]	-	4	-
ILremver	[MTAF]	-	4	-
ICpla	[MTAF]	-	4	-
ICdc	[MTAF]	-	4	-
IChor	[MTAF]	-	4	-
ICver	[MTAF]	-	4	-
ICang2	[MTAF]	-	4	-
ICang18	[MTAF]	-	4	-
ICang34	[MTAF]	-	4	-
ICremhor	[MTAF]	-	4	-
ICremver	[MTAF]	-	4	-
ILreffilt	[MTAF]	-	4	-
ILrefnfilt	[MTAF]	-	4	-
InoMPM	[HSR+]	-	1	-

Tabelle 6.1: Referenzen der Features und Indizes, welches Feature in welchem Model verwendet wird. Die Nummer gibt an, wie viele Blockgrößen berüksichtigt werden (bei einer 1 werden Blockgrößen nicht berücksichtigt).

Index	Referenz	BF27	BFcomplete	BFHL
PLinter	[HWD$^+$]	-	4	-
PLinterres	[HSR$^+$]	4	4	4
PLinterMV	[HSR$^+$]	-	11	-
PLmerge	[HSR$^+$]	-	11	-
PLskip	[HSR$^+$]	4	4	4
MC	[SA]	-	1	-
PLfrac	[HSR$^+$]	1	1	1
PLfracHor	[HSR$^+$]	-	4	-
PLfracVer	[HSR$^+$]	-	4	-
PCHpel	[HSR$^+$]	-	4	-
MVD	[HSR$^+$]	-	1	-
bi	[HSR$^+$]	1	1	1
c	[HSR$^+$]	1	1	1
v	[HWD$^+$]	1	1	1
CSBF	[HSR$^+$]	-	1	-
cgl	[HSR$^+$]	-	1	-
Tr	[HSR$^+$]	4	4	4
LITr	[HSR$^+$]	-	4	-
CITr	[HSR$^+$]	-	4	-
LPTr	[HSR$^+$]	-	4	-
CPTr	[HSR$^+$]	-	4	-
TS	[HSR$^+$]	-	1	-
B	[HSR$^+$]	1	1	1
Bs0	[HSR$^+$]	-	1	-
Bs1	[HSR$^+$]	-	1	-
Bsg0	[SA]	-	1	-
Lsao	[HWD$^+$]	1	1	1
Csao	[HSR$^+$]	1	1	1
LsaoBO	[HSR$^+$]	-	1	-
CsaoBO	[HSR$^+$]	-	1	-
LsaoEO	[HSR$^+$]	-	1	-
CsaoEO	[HSR$^+$]	-	1	-
saoAll	[HSR$^+$]	-	1	-
width	[HSR$^+$]	-	-	1
height	[HSR$^+$]	-	-	1
NrFrames	[HSR$^+$]	-	-	1
size	[HSR$^+$]	-	-	1

Tabelle 6.2: Referenzen der Features und Indizes, welches Feature in welchem Model verwendet wird. Fortsetzung der Tabelle 6.1.

QCIF (176 x 144 pixel)	CIF (352 x 288 pixel)
Akiyo (30 frames)	Foreman (30 frames)
Crew (50 frames)	Tennis (30 frames)
Miss America (50 frames)	Car Phone (50 Frames)
Coastguard (50 Frames)	Bus (50 Frames)
News (30 frames)	Suzie (30 frames)

Tabelle 6.3: Evaluierungssequenzen in CIF und QCIF Format [uni]

Klasse A (2560 x 1600 pixel)	Klasse B (1920 x 1080 pixel)
PeopleOnStreet	BasketballDrive
Traffic	BQTerrace
	Cactus
	Kimono
	ParkScene
Klasse C (832 x 480 pixel)	**Klasse D (416 x 240 pixel)**
BasketballDrill	BasketballPass
BQMall	BlowingBubbles
PartyScene	BQSquare
RaceHorses	RaceHorses
Klasse E (1280 x 720 pixel)	**Klasse F (variable Auflösung)**
FourPeople	SlideEditing
Johnny	SlideShow
KristenAndSara	ChinaSpeed
BasketballDrillText	

Tabelle 6.4: Evaluierungsequenzen aus [Bos]. Außer Klasse A (8 frames) und Klasse E (500 frames) wurden alle Sequencen mit 40 frames codiert.