technische universität
dortmund

Bachelorarbeit

Optische Erkennung von Dudelsacknoten

Andreas Jöbges
27. Mai 2017

Betreuer: Prof. Dr. Günter Rudolph
Zweitbetreuer Dr. Igor Vatolkin

Fakultät für Informatik
Algorithm Engineering (Ls11)
Technische Universität Dortmund
http://ls11-www.cs.tu-dortmund.de

Inhaltsverzeichnis

Kapitel 1

Einleitung

Für das Problem der optischen Erkennung von Noten (auch „OMR“ für „Optical Music Recognition“ genannt) für den Dudelsack gibt es bisher keine Lösungsansätze. In dieser Arbeit wird der Versuch beschrieben, die Umwandlung von geschriebenem Notenmaterial für Dudelsack in ein Format vorzunehmen, das mit Computer bearbeitet werden kann. Der hier verwendete Ansatz weicht dabei sowohl bei der Segmentierung als auch bei der Klassifizierung der erhaltenen Einzelsymbole vom klassischen Vorgehen in der OMR ab.

In den folgenden Kapiteln wird der für die Arbeit programmierte Algorithmus beschrieben. Danach folgen einige Tests, die zum einen zeigen, daß der Ansatz prinzipiell geeignet ist, das Problem zu lösen, zum anderen aber auch seine Schwächen aufzeigen. Abschließend werden mögliche Lösungswege aufgezeigt.

1.1 Motivation und Hintergrund

Optische Notenerkennung ist ein bekanntes Problem, zu dem die erste bekannte Veröffentlichung von Dennis Pruslin 1966 [11] erschienen ist. Pruslin beschreibt darin ein Programm zur Notenerkennung, das allerdings nur stark eingeschränkte Merkmale erkennen kann. So werden nur Noten mit gefüllten Köpfen und in beschränktem Umfang Akkorde erkannt. Alle anderen Zeichen, die zum Schreiben komplexerer Musik nötig sind, kann sein Programm nicht verarbeiten [8].

Homenda/Luckner [7] führten 2004 Tests zur Erkennung von Musiksymbolen unter anderem mit neuronalen Netzen durch. Allerdings wurde ein sehr eingeschränkter Satz an Symbolen verwendet, der im wesentlichen aus Pausen-, Piano- und Fortesymbolen bestand. Es zeigte sich, daß neuronale Netze hier gute Ergebnisse lieferten.

Zum Zeitpunkt der Erstellung der vorliegenden Arbeit existieren einige kommerzielle Produkte, mit denen Noten gescannt und für den Computer lesbar gemacht werden können. Zu den namhaftesten zählen hier: SmartScore[1], PhotoScore[2], capella-scan[3] und SharpEye Music Reader[4].

Ein Notenwerk, das von einer OMR verabeitet wurde, kann anschließend auf vielfältige Weise weiterbenutzt werden. So können Teile verändert, neu formatiert und gedruckt werden. Es kann eine MIDI-Datei[5] erstellt werden, um das Stück per Computer abzuspielen etc.

Zum Problem der optischen Erkennung von Dudelsacknoten sind bisher keine Arbeiten und Anwendungen bekannt. Eine diesbezügliche Internetrecherche bei der Literaturdatenbank „Scopus“[6] lieferte keine Ergebnisse.

Natürlich kann man für einen Scan von Dudelsacknoten auch die bekannten Produkte verwenden, diese sind aber nicht an die Eigenheiten geschriebener Dudelsackmusik angepaßt. Die Unterschiede liegen hier hauptsächlich in der großen Zahl an standardisierten Vorschlagsnoten („Grace Notes“) und der Tatsache, daß die Notenhälse aller Melodienoten nach unten gerichtet sind. Die Notenhälse lassen sich vermutlich per Einstellung verändern, aber die Verzierungen nur mit großem Aufwand, da jede individuell konfiguriert werden muß. Andererseits gibt es funktionsfähige Produkte, mit denen Dudelsacknoten in der gewünschten Form dargestellt werden können. Eine Möglichkeit, einen Scan in eines dieser Formate umzuwandeln, existiert zur Zeit nicht. Hier setzt die vorliegende Arbeit an und überführt eine graphische Darstellung von Dudelsacknoten in das Format bww.

Die spezifischen Eigenheiten geschriebener Dudelsackmusik sind einerseits mit vorhandenen Anwendungen nur schwer wiederzugeben, andererseits stellen diese Produkte einen großen Overhead an Optionen zur Verfügung, die für die Umwandlung von Dudelsacknoten nicht benötigt werden. So gibt es keine Akkorde oder parallel verlaufenden Melodielinien. Die Musik ist monophon und nur Überbindungen und Voltenklammern sind über mehrere Noten hinweg relevant. Dies führt wiederum zu einer Vereinfachung des Problems.

[1] www.musitek.com, aufgerufen am 19.Mai 2017
[2] www.neuratron.com, aufgerufen am 19.Mai 2017
[3] www.capella-software.com, aufgerufen am 19.Mai 2017
[4] www.visiv.co.uk, aufgerufen am 19.Mai 2017
[5] www.midi.org aufgerufen am 19.Mai 2017
[6] www.scopus.com, aufgerufen am 2.Mai 2017

1.2 Aufbau der Arbeit

In Kapitel 2 werden die für das Verständnis der Arbeit nötigen Begriffe geklärt. Nach einer kurzen Definition des Begriffes „Optical Music Recognition“ wird die grundlegende Funktionsweise eines neuronalen Netzes erklärt, welches den Kern des Erkennungsmechanismus der hier beschriebenen OMR bildet. Hierzu wird zunächst die Funktionsweise eines Neurons erklärt, um dann Aufbau, Funktion und Training eines neuronalen Netzes anzureißen.

Im Anschluß daran wird Aufbau und Syntax einer bww-Datei erklärt und eine Beispieldatei vorgestellt. Den Abschluß von Kapitel 2 bildet eine kurze Erklärung der Funktionsweise des schottischen Dudelsacks.

Kapitel 3 beschäftigt sich mit dem für diese Arbeit entwickelten Algorithmus zur OMR. Zunächst wird der Algorithmus in groben Zügen dargestellt um dann in Abschnitt 3.2 detailliert vorgestellt zu werden.

In Kapitel 4 soll eine Bewertung des vorliegenden Algorithmus vorgenommen werden. Hierfür werden zunächst die Rahmenbedingungen für die Versuche abgesteckt und eine Metrik eingeführt, die es ermöglicht, die Resultate quantitativ zu fassen.

Mit Hilfe dieser Metrik wird dann untersucht, ob der Ansatz, die Notenlinien nicht zu entfernen, vergleichbare Ergebnisse zum klassischen Ansatz liefert, in dem zuerst die Notenlinien entfernt werden.

Ein weiteres Experiment klärt, ob das Hinzufügen von künstlich erzeugten Trainingsdaten eine Verbesserung der Erkennungsrate mit sich bringt.

Zuletzt wird untersucht, an welcher Stelle die wesentlichen Erkennungsfehler gemacht werden und womit dies zusammenhängt.

Abschließend werden in Kapitel 5 die Ergebnisse der Arbeit zusammengefaßt und ein Ausblick auf mögliche Weiterentwicklungen der hier getesteten OMR gegeben. Außerdem werden mögliche weitere Experimente, die damit durchgeführt werden können, diskutiert.

1.3 Begriffsbestimmung

In der gesamten Arbeit bezieht sich der Begriff „Dudelsack“ auf den schottischen Dudelsack und die für dieses Instrument üblicherweise verwendete Notation.

Das hier verwendete Programm zum Notensatz nennt sich „Bagpipe Player“ [14]. Das dazugehörige Format zur Notation und Speicherung hat die Dateiendung „bww“.

Die verwendete Programmiersprache ist MATLAB R2016b.

Üblicherweise bezeichnen die Begriffe Notensystem, Notenzeile und Notenlinie synonym ein System, bestehend aus fünf Linien zur Notation von Musik. Da in dieser Arbeit aber zwischen dem System aus fünf Linien und der einzelnen Linie unterschieden werden muß, soll hier der Begriff Notensystem ein System aus fünf Notenlinien bezeichnen, wobei eine Notenlinie aus einem einzelnen Geradenabschnitt besteht.

Kapitel 2

Grundlagen und Methoden

2.1 Optical Music Recognition

Das am weitesten verbreitete System, die wesentlichen Elemente von Musik (Tonhöhe, -dauer und -lautstärke) graphisch festzuhalten, ist die moderne westliche Notenschrift.

Mit der fortschreitenden Digitalisierung der Welt entsteht der Wunsch, diese Notation auch in digitaler Form zugänglich zu machen. Für die Schriftsprache wird das „OCR - Optical Character Recognition“ genannte Verfahren dazu genutzt, gedruckte oder handschriftliche Schriftstücke zunächst zu scannen und dann durch automatisierte Schrifterkennung von einer reinen Bilddatei in eine Textdatei zu überführen.

Auch für musikalische Schriften gibt es digitale Notationen wie zum Beispiel „musicXML“[1] oder „abc“[2]. Für die vorliegende Arbeit wird das zu dem Programm Bagpipe Player gehörende Format bww genutzt. Die Überführung von musikalischen Schriftstücken, die in Bildform vorliegen, in eine digitale Musiknotation wird OMR, zu deutsch: Optische Musik- oder Notenerkennung, genannt.

Die klassische Vorgehensweise in diesem Prozeß besteht dabei aus folgenden Schritten [2]:

1. Separierung der Notensysteme
2. Erkennen und Entfernen der Notenlinien
3. Segmentierung der Notensysteme in „Primitivsymbole“ wie zum Beispiel Ovale (Notenköpfe), Quader (Notenhälse, Balken) etc.

[1] www.musicxml.com, aufgerufen am 19.Mai 2017
[2] www.abcnotation.com, aufgerufen am 19.Mai 2017

4. Klassifizierung der in Schritt 3 erkannten Objekte

5. Semantisches Zusammenfügen der erhaltenen Elemente

Ein weiterer Überblick über die Methoden und Schwierigkeiten von OMR-Systemen findet sich in [5]. Eine ausführliche Diskussion eines weitestgehend klassischen OMR-Systems findet sich bei [13]. Der zweite Teil dieser Arbeit beschäftigt sich ausführlich mit der Möglichkeit, Fehler im Rahmen der Nachbearbeitung zu korrigieren.

2.2 Neuronale Netze

Zur Segmentierung eines Notenblatts in einzelne Notenelemente werden heuristische Methoden wie das Zählen von Pixeln in einer Reihe/Spalte oder das Finden von zusammenhängenden Strukturen durch Betrachtung der benachbarten Pixel angewendet. Hierauf wird im einzelnen in Kapitel 3.2 eingegangen. Um die so segmentierten Symbole zu klassifizieren wird hier ein neuronales Netz verwendet.

2.2.1 Neuronen

Ein neuronales Netz besteht aus Neuronen, die parallel oder hintereinander geschaltet werden. Jedes einzelne Neuron (Abbildung 2.1) ist in den meisten Fällen eine Funktion, die als Eingabe einen n-dimensionalen reellwertigen oder binären Vektor erhält und als Ausgabe einen Wert zwischen 0 und 1 erzeugt: $N : \mathbb{R}^n \to [0, 1]$. Dies geschieht üblicherweise in zwei Schritten. Zur Vorbereitung wird der Vektor um eine Dimension erweitert, die immer den Wert 1 hat $(x_1...x_n) \to (1, x_1...x_n)$. Dies geschieht, um die Geradengleichung $y = m_0 + m_1 \cdot x_1 + \cdots m_n \cdot x_n$ in eine Form zu bringen, die eine Vektormultiplikation unterstützt:

$$y = m_0 \cdot 1 + m_1 \cdot x_1 + \cdots m_n \cdot x_n = (m_0 \ \cdots \ m_n) \cdot (1 \ x_1 \cdots x_n)^T$$

Im ersten Schritt wird dieser Eingabevektor $\vec{x} = (1 \ x_1 \ \cdots \ x_n)$ durch eine lineare Abbildung

$$f(x) = \vec{m} \cdot \vec{x}^T = (m_0 \ m_1 \ \cdots \ m_n) \cdot (1 \ x_1 \ \cdots \ x_n)^T = \sum_{i=0}^{n} m_i x_i$$

auf einen einzelnen Wert abgebildet.

Im zweiten Schritt wird das Ergebnis des ersten Schrittes dann auf einen Wert zwischen 0 und 1 abgebildet. Dies kann auf unterschiedliche Weise geschehen. Die Einfachste setzt alle Werte größer oder gleich einem gegebenen Schwellenwert t auf

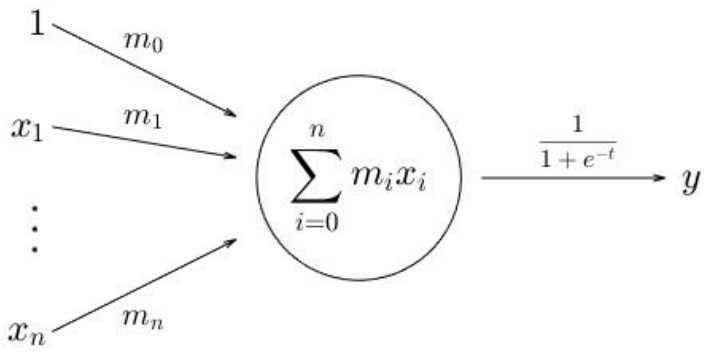


Abbildung 2.1: Einzelnes Neuron

1 und alle Werte kleiner als t auf 0, oder im umgekehrten Fall alle Werte kleiner oder gleich t auf 1 und alle größer t auf 0. Hierbei wird jedoch nicht der ganze Wertebereich [0,1] abgedeckt. Außerdem ist diese Funktion weder stetig noch differenzierbar. Diese Eigenschaften werden aber später für die Anpassung der Gewichte durch den sogenannten „Backpropagation Algorithmus“ benötigt. Um dies zu ermöglichen, verwendet man häufig die Schwanenhalsfunktion („sigmoid function“) $g(t) = \frac{1}{1+e^{-t}}$.

Die Anwendung dieser Funktion kann man so interpretieren, daß jedem n-dimensionalen Vektor $\vec{v}$ ein Grad von Zugehörigkeit zu einer Menge M zugewiesen wird, wobei Vektoren mit $N(\vec{v}) = 0$ mit Sicherheit nicht in der Menge liegen und solche mit $N(\vec{v}) = 1$ sicher Elemente von M sind. Abbildung 2.1 zeigt eine graphische Darstellung eins solchen einzelnen Neurons.

2.2.1 Lemma. *Alle Vektoren mit gleichem Funktionswert liegen auf der gleichen Hyperebene.*

Beweis. Sei eine Ebene gegeben durch die drei Vektoren x, y und z mit $f(x) = f(y) = f(z) = k$ und $f(x) = \sum_{i=0}^{n} m_i x_i$. Dann spannt $E(a,b) = x + a \cdot (y - x) + b \cdot (z - x)$ eine Hyperebene auf. Für alle Vektoren $E(a,b)$ dieser Ebene gilt dann:

$$\begin{aligned} f(E(a,b)) &= \sum_{i=0}^{n} m_i(x_i + a(y_i - x_i) + b(z_i - x_i)) \\ &= \sum_{i=0}^{n} m_i x_i + a\sum_{i=0}^{n} m_i y_i - a\sum_{i=0}^{n} m_i x_i + b\sum_{i=0}^{n} m_i z_i - b\sum_{i=0}^{n} m_i x_i \\ &= f(x) + af(y) - af(x) + bf(z) - bf(x) \\ &= k + ak - ak + bk - bk \\ &= k \end{aligned}$$

□

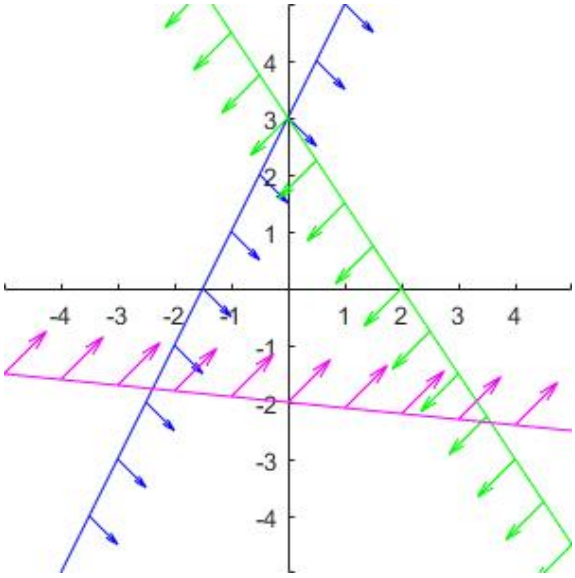


Abbildung 2.2: Von drei Neuronen erzeugte dreieckige Teilmenge des $\mathbb{R}^2$

Wenn nun $M = \{\vec{v} \in \mathbb{R}^n | f(\vec{v}) >= 0,5\}$ ein Halbraum des $\mathbb{R}^n$ ist, so ist leicht zu sehen, daß die Ebene $E = \{\vec{v} \in \mathbb{R}^n | f(\vec{v}) = 0,5\}$ den $\mathbb{R}^n$ in M und $\mathbb{R}^n \backslash M$ teilt.

2.2.2 Neuronales Netz

Es ist leicht zu erkennen, daß durch eine Und-Verknüpfung mehrerer Neuronen jede beliebige konvexe Menge approximiert werden kann (Abbildung 2.2 zeigt eine von drei Neuronen erzeugte zweidimensionale Menge). In einem zweiten Schritt kann man dann durch die Vereinigung dieser Mengen jede beliebige Menge approximieren.

Ein neuronales Netz besteht aus mehreren Schichten, die ihrerseits wiederum aus mehreren Neuronen bestehen (Abbildung 2.3 zeigt ein einfaches neuronales Netz aus drei Schichten mit je drei Neuronen in den Schichten 1 und 2 und einem Neuron in Schicht 3). Der Eingabevektor $\vec{x}$ ist hierbei die Eingabe für alle Neuronen der ersten Schicht. Alle Ausgaben der Neuronen einer Schicht erzeugen dann den Eingabevektor für die nächste Schicht. Die letzte Schicht erzeugt die Ausgabe, wobei auch mehrdimensionale Ausgaben möglich sind, wenn die letzte Schicht mehrere Neuronen enthält.

Die Gewichte jeder Schicht W_i sind hierbei Matrizen. Die Ausgabe der jeweiligen Schicht berechnet sich dann durch einfache Multiplikation des Ausgabevektors der vorhergehenden Schicht mit der Gewichtsmatrix und anschließende Anpassung jedes Wertes mit Hilfe der Schwanenhalsfunktion.

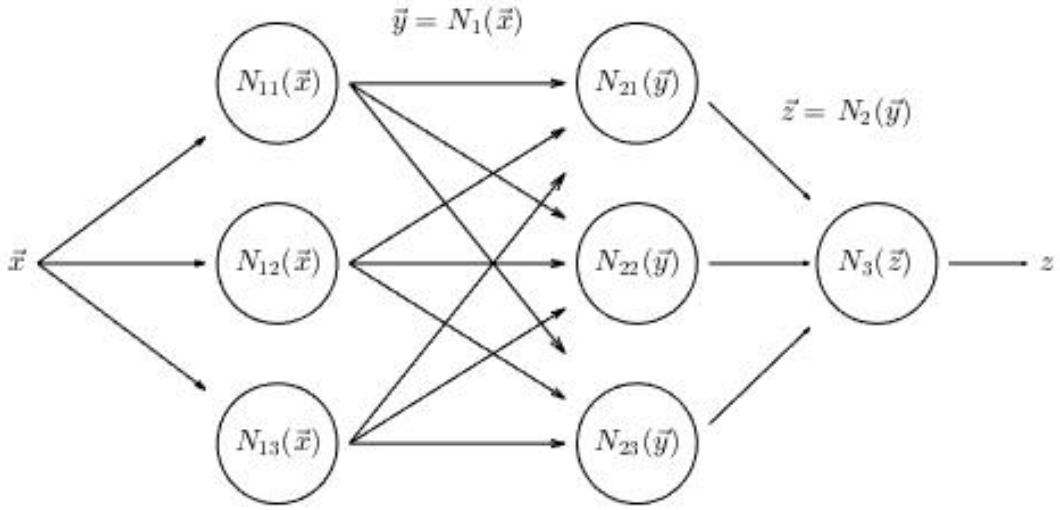


Abbildung 2.3: Einfaches neuronales Netz

Neuronale Netze werden unter anderem verwendet, um Eingaben zu klassifizieren. Eine eindimensionale Ausgabe gibt dann die Konfidenz aus, mit der die Eingabe zu der Menge gehört. Um mehrere Mengen zu separieren, gibt man für jede Menge eine Konfidenz aus (je ein Ausgabeneuron) und wählt die Menge mit der höchsten Konfidenz.

Es bleibt noch das Problem, die Gewichtsvektoren m für jedes Neuron zu ermitteln. Neuronale Netze gehören zu den überwachten Lernalgorithmen. Diese funktionieren, indem Trainingsbeispiele mit vorgegebenen Klassifizierungen gegeben sind und der Algorithmus (hier das neuronale Netz) so lange angepaßt wird, bis er die Beispiele oder zumindest einen möglichst großen Teil davon richtig klassifiziert. Zum Ermitteln der Gewichtsvektoren $\vec{m}$ bedient man sich des sogenannten „Backpropagation-Algorithmus“. Hierzu wird zunächst das neuronale Netz mit zufälligen Gewichten initialisiert. Dann werden für alle Trainingsbeispiele die Ausgabewerte des Netzes ermittelt und mit den erwarteten Werten verglichen. Nun kann man mit Hilfe der partiellen Ableitungen nach den einzelnen Komponenten des Eingabevektors (für das Ausgabeneuron in Abbildung 2.3 wäre dies $\vec{z} = N_2(\vec{y})$) die Gewichte mit Hilfe des Gradientenabstiegs anpassen und ermitteln, wie groß die Fehler in der Schicht davor waren. Dieser Prozeß setzt sich dann rückwärts durch das neuronale Netz fort. Nachdem das gesamte Netz angepaßt wurde, führt man die nächste Iteration des Backpropagation-Algorithmus aus, bis das Ergebnis den gegebenen Anforderungen entspricht. Um eine Überanpassung des Netzes an die Trainingsbeispiele zu verhindern führt man einen Prozeß namens „Crossvalidation“

durch. Hierzu teilt man üblicherweise die Trainingsmenge in n gleichgroße Teile auf (z.B. 10%). Dann wählt man einen Teil zur späteren Validierung aus und trainiert mit den restlichen Teilen das neuronale Netz. Mit dem zur Validierung gewählten Teil ermittelt man nun die Zahl der Fehler. Nachdem man jede der n Teilmengen einmal zur Validierung benutzt hat, wählt man das Netz, welches am wenigsten Fehler gemacht hat.

Für weitere Informationen zu neuronalen Netzen siehe z.B. [12].

2.3 Dudelsacknotation mit dem Bagpipe Player

Die Notation von Dudelsackmusik weist ein paar Besonderheiten im Vergleich zur üblichen Notation von Musik auf. Der Dudelsack ist ein monophones Instrument. Daher sind keine Akkorde möglich. Der Notenumfang reicht vom g' bis zum a". Die Tonleiter ist mixolydisch mit a' als Grundton. In der Notation werden alle Notenhälse nach unten geschrieben. Der größte Unterschied zur üblichen Notation ist eine umfangreiche Sammlung an Grace Notes. Grace Notes sind standardisierte Verzierungen, die aus einzelnen Noten oder aus Gruppen von bis zu sieben Noten bestehen. Diese Grace Notes werden mit kleineren Notenköpfen und mit den Hälsen nach oben dargestellt. Sie werden bis auf wenige Ausnahmen als Zweiunddreißigstel geschrieben und mit Balken gruppiert. Desweiteren gibt es einige Konventionen zur Vereinfachung, wie zum Beispiel das Weglassen des Wiederholungszeichens am Anfang eines Teilstücks/„Parts“ oder der beiden Kreuze zum Bestimmen der Tonart. Im ersten Fall ist per Konvention klar, wo die Wiederholung beginnt, und da die Tonart beim Dudelsack nicht veränderbar ist, kann auf die Darstellung der Kreuze verzichtet werden. Überbindungen kommen in der Dudelsacknotation nur zur Verbindung zweier gleicher Noten oder zur Markierung von Triolen, niemals aber als Legatobogen vor, da der Dudelsack aufgrund seiner Bauart nur legato spielen kann.

Die Codierung im Bagpipe Player ordnet jeder Note, Verzierung und sonstigen Symbolen im Notensystem eine Gruppe von ASCII-Zeichen zu. Eine Melodienote wird beispielsweise durch den entsprechenden Großbuchstaben codiert. Um g' von g" und a' von a" zu unterscheiden, werden LG (Low G) bzw. HG (High G) und LA bzw. HA verwendet. Dem Buchstaben folgt ein Unterstrich und dann der Nenner der Notenlänge als Zahl. Eine Viertelnote C wird dementsprechend durch „C_4“ dargestellt. Jedes Symbol wird durch ein Leerzeichen oder einen Tabulator vom nächsten getrennt. Um mehrere Noten zu gruppieren, indem man ihre Fähnchen als Balken verbindet, schreibt man direkt hinter der Notenbezeichnung ein „r“ oder

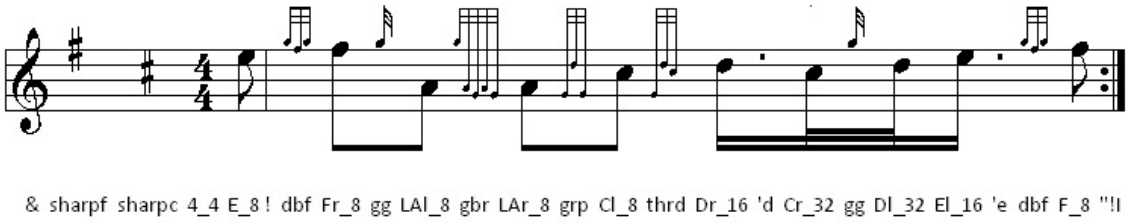


Abbildung 2.4: Einige Notenbeispiele mit Kodierung im „bww-Format“

„l“. Außerdem werden Notengruppen so lange miteinander verbunden, bis sie durch einen Tabulator unterbrochen werden.

Verzierungen werden durch Buchstabenfolgen kodiert, die meist Abkürzungen der Namen sind. So bezeichnet beispielsweise „dbx“ eine Verzierung namens „Doubling“ (Abbildung 4.9), wobei das „x“ für die zugehörige Note steht. Ein Doubling auf E wäre dann „dbe“. Einige Notierungsbeispiele finden sich in Abbildung 2.4.

Ein Notensystem startet mit dem Violinschlüssel („&“), gefolgt von zwei Kreuzen („sharpf sharpc “). In der ersten Zeile folgt noch das Metrum (hier „4_4“). Eine Notensystem endet mit „!t“(Zeilenende) oder „!I“ (Ende eines Teilstücks/Parts). Einzelne Elemente werden durch Leerzeichen getrennt, Notengruppen durch Tabulatoren, Takte durch einen Taktstrich („!“).

Eine vollständige Liste aller Verzierungen und der zugehörigen Abkürzungen findet sich in der Dokumentation des Bagpipe Player [14].

Eine bww-Datei beginnt mit einem Header, in dem Informationen zur Formatierung des Druckbilds, der Spielgeschwindigkeit (Taktschläge pro Minute), der Geschwindigkeit einzelner Grace Notes und der Frequenzen aller Tonhöhen gespeichert werden. Nach dem Header folgen Titel und andere Textelemente und abschließend die einzelnen Notensysteme.

Im Folgenden nun ein Beispiel für eine vollständige bww-Datei und das daraus erzeugte Notenbild. Das Stück heißt „The Green Hills of Tyrol“. Es stammt ursprünglich aus der Oper „Willhelm Tell“ von Rossini und wurde von Pipe-Major John MacLeod während des Krimkriegs 1854-1856 für den Dudelsack adaptiert [4].

Bagpipe Reader:1.0

MIDINoteMappings,(54,56,58,59,61,63,64,66,68,56,58,60,61,63,65,66,68,70,55,57,59,60,62,64,65,67,69)
FrequencyMappings,(370,415,466,494,554,622,659,740,831,415,466,523,554,622,699,740,831,932,392,440,494,523,587,659,699,784,880)

InstrumentMappings,(71,71,45,33,1000,60,70)
GracenoteDurations,(45,60,30,50,100,200,800,1200,250,250,250,500,200)
FontSizes,(100,100,100,100,250)
TuneTempo,90
TuneFormat,(1,0,F,L,500,500,500,500,L,1,0)

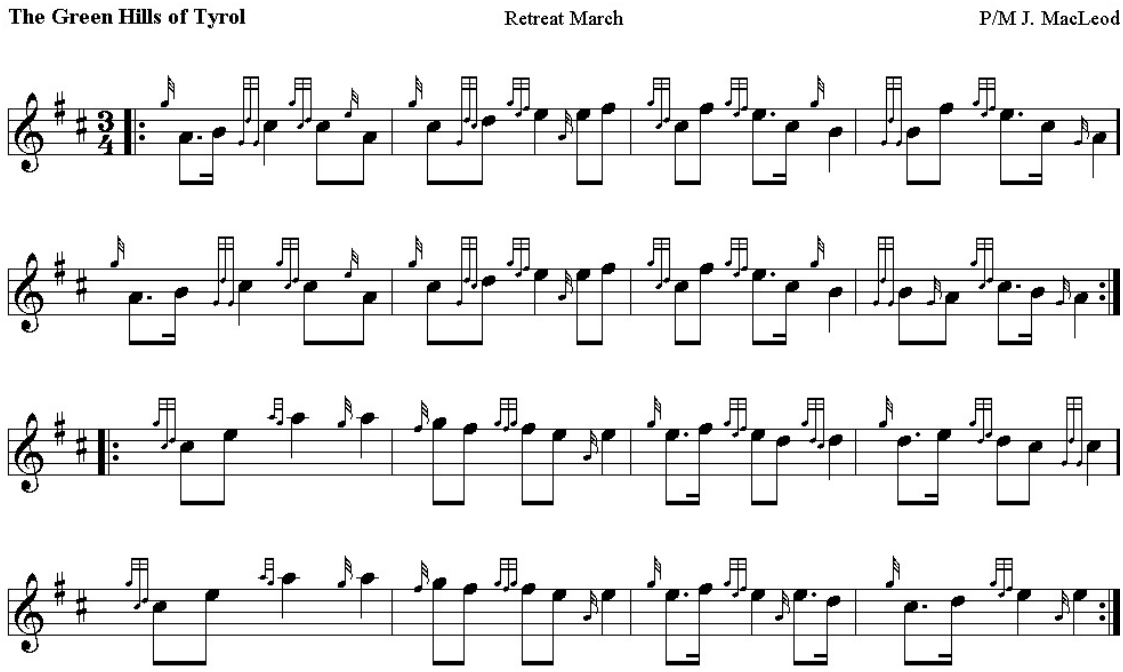
"The Green Hills of Tyrol",(T,L,0,0,Times New Roman,16,700,0,0,18,0,0,0)
"Retreat March",(Y,C,0,0,Times New Roman,14,400,0,0,18,0,0,0)
"P/M J. MacLeod",(M,R,0,0,Times New Roman,14,400,0,0,18,0,0,0)

& sharpf sharpc 3_4 *I!"*
gg LAr_8 'la Bl_16 *grp C_4* *dbc Cr_8 eg LAl_8* *!*
gg Cr_8 thrd Dl_8 *dbe E_4* *strla Er_8 Fl_8* *!*
dbc Cr_8 Fl_8 *dbe Er_8 'e Cl_16* *gg B_4* *!*
grp Br_8 Fl_8 *dbe Er_8 'e Cl_16* *strlg LA_4* *!t*

& sharpf sharpc
gg LAr_8 'la Bl_16 *grp C_4* *dbc Cr_8 eg LAl_8* *!*
gg Cr_8 thrd Dl_8 *dbe E_4* *strla Er_8 Fl_8* *!*
dbc Cr_8 Fl_8 *dbe Er_8 'e Cl_16* *gg B_4* *!*
grp Br_8 strlg LAl_8 *dbc Cr_8 'c Bl_16* *strlg LA_4* *"!I*

& sharpf sharpc *I!"*
dbc Cr_8 El_8 *dbha HA_4* *strhg HA_4* *!*
strf HGr_8 Fl_8 *dbf Fr_8 El_8* *strla E_4* *!*
gg Er_8 'e Fl_16 *dbe Er_8 Dl_8* *lgstd D_4* *!*
gg Dr_8 'd El_16 *dbd Dr_8 Cl_8* *grp C_4* *!t*

& sharpf sharpc
dbc Cr_8 El_8 *dbha HA_4* *strhg HA_4* *!*
strf HGr_8 Fl_8 *dbf Fr_8 El_8* *strla E_4* *!*
gg Er_8 'e Fl_16 *dbe E_4* *strla Er_8 'e Dl_16* *!*
gg Cr_8 'c Dl_16 *dbe E_4* *strla E_4* *"!I*



2.4 Der schottische Dudelsack

Abbildung 2.5 zeigt eine schematische Darstellung des schottischen Dudelsacks. Der genaue englische Begriff lautet „Great Highland Bagpipes“. Es handelt sich um ein mundgeblasenes Instrument. Der Spieler (englisch: „Piper“) befüllt mit Hilfe der „Blowpipe“, eines Rohres mit Rückschlagventil, einen Luftsack („Pipe bag“), der als Vorratsbehälter dient. Durch Zusammendrücken mit dem Arm hält der Piper den Druck in diesem Luftsack konstant, während er ihn mit dem Blasrohr regelmäßig nachfüllt. Durch die Luft im Sack werden die Pfeifen mit einem konstanten Luftstrom versorgt. Diese Pfeifen sind zum einen die Spielpfeife („Chanter“), die einer Oboe ähnlich durch ein Doppelrohrblatt („Reed“) den Ton erzeugt. Der Chanter hat acht Grifflöcher zum Spielen der Melodie. Die drei über der Schulter getragenen Bordunpfeifen (zwei „Tenordrones“ und eine „Bassdrone“) sind ein bzw. zwei Oktaven tiefer als der Grundton der Spielpfeife gestimmt. Sie erzeugen einen konstanten Hintergrundton durch einfache Rohrblätter.

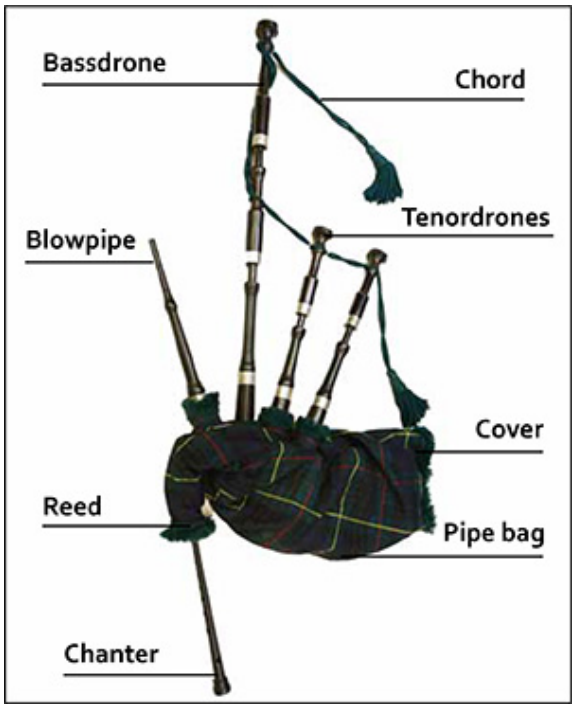


Abbildung 2.5: Schematische Darstellung eines Dudelsacks mit Bezeichnung der wesentlichen Bauteile [10]

Kapitel 3

Der Algorithmus

3.1 Grundlegende Vorgehensweise

Der erste Schritt beim Verarbeiten eines Notenblatts ist die vertikale Separierung in einzelne Notensysteme. Dies kann dadurch geschehen, daß man die schwarzen Pixel in jeder Reihe zählt (Abbildung 3.1). Hierbei kann man deutlich die einzelnen Notensysteme erkennen. Die Reihen zwischen den Systemen enthalten keine schwarzen Pixel. Die Segmentierung fällt hier leicht, da man alle Zeilen mit 0 schwarzen Pixeln entfernen kann. Die jeweils dazwischen liegenden Reihen sind dann die Notensysteme (der Titel wird hier auch als Zeile erkannt und später entfernt).

Als nächstes gilt es, die erhaltenen Notensysteme in einzelne Symbole zu segmentieren und diese zu klassifizieren. Das traditionelle Vorgehen ist, die Notenlinien zu entfernen und zusammenhängende schwarze Flächen zu separieren. Danach sucht man nach primitiven Elementen wie Ellipsen (Notenköpfe), Quadern (Notenhälse, Balken, Taktstriche), etc. Diese werden dann analysiert und wieder zu musikalischen Symbolen synthetisiert [2].

Dieses Verfahren hat allerdings den Nachteil, daß schlecht gescannte Noten oder Notenbilder, die vom Standard stark abweichen (z.B. auch handschriftliche Noten) die Erkennung deutlich erschweren. Die Separierung nach Einzelelementen bringt allerdings auch Probleme mit sich. Hierbei können z.B. einzelne Elemente zerteilt oder mehrere Einzelelemente als eines erkannt werden.

Der in der vorliegenden Arbeit verfolgte Ansatz segmentiert ein Notensystem in die einzelnen musikalischen Symbole mit einer leichten Variation des Algorithmus zur Segmentierung der Notensysteme, um diese dann durch ein neuronales Netz erkennen zu lassen. Dadurch wird das Problem auf die saubere Trennung der einzelnen Symbole und das Training des neuronalen Netzes reduziert. Dies hat den Vorteil, daß

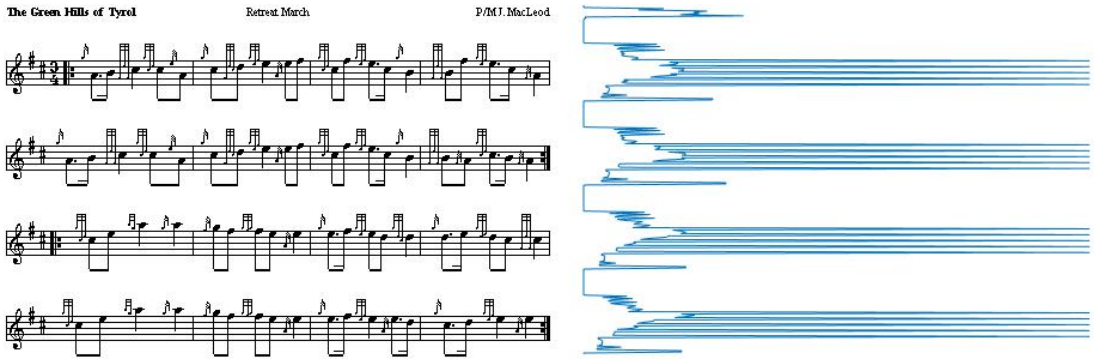


Abbildung 3.1: Vergleich eines Notenblattes mit der Zahl der Pixel in jeder Reihe.

für ein neues Druckbild oder eine neue Handschrift nur das neuronale Netz nachtrainiert werden muß, ohne den eigentlichen Algorithmus verändern zu müssen. Hierzu ist anzumerken, daß das Erkennen handschriftlicher Noten nicht Teil dieser Arbeit ist, sondern eine mögliche Weiterentwicklung des Programms darstellt. Neuronale Netze lernen beim Training, die Eingabe einer der vorgegebenen Ausgabemöglichkeiten zuzuordnen. In diesem Fall wird also die Abbildung eines Notensymbols auf ein Symbol aus dem Alphabet des Bagpipe Players abgebildet.

Das als Beispiel verwendete Stück (Abbildung 3.1) wurde direkt aus einem Notensatzprogramm heraus als Grafik erzeugt und liefert hierbei gute Ergebnisse.

Bei der dann folgenden horizontalen Segmentierung eines Notensystems entstehen im erzeugten Histogramm keine Nullinien (wie dies bei der Aufteilung in Notensysteme z.B. in Abbildung 3.1 der Fall ist), da hier immer mindestens die Notenlinien als schwarze Pixel entstehen. Dementsprechend wird hier der Schwellenwert größer als 0 gewählt. Der Begriff Histogramm wird in diesem Zusammenhang für die graphische Darstellung einer Liste von Werten in Form eines Balkendiagramms verwendet, wobei jeder Wert einen eigenen Balken hat.

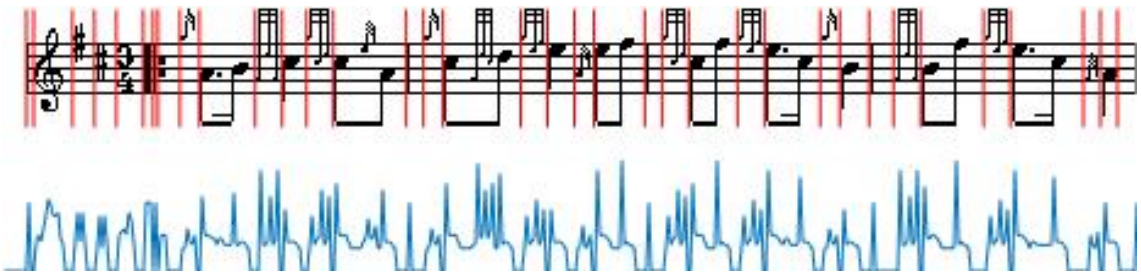


Abbildung 3.2: Vergleich eines Notensystems mit der Zahl der Pixel in jeder Spalte und den daraus resultierenden Schnitten

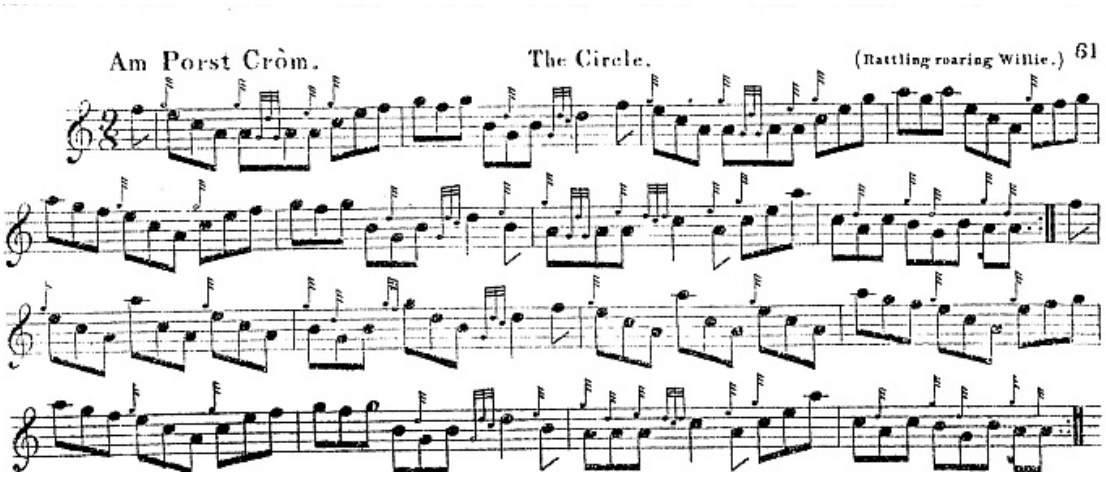


Abbildung 3.3: Beispiel für vertikal zu dicht gesetzte Notensysteme [6]

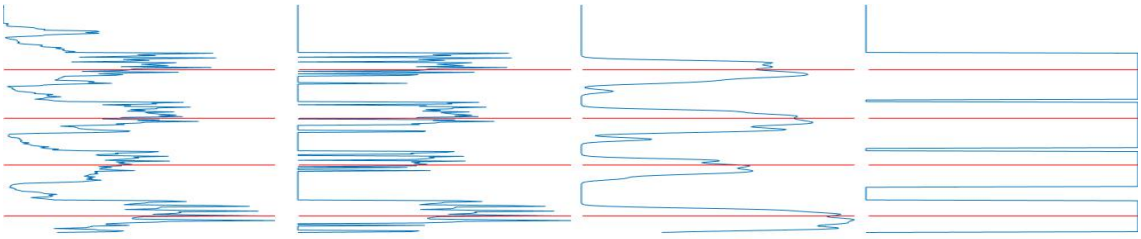


Abbildung 3.4: Schätzung der Lage der Notensysteme zum Stück in Abbildung 3.3

Abbildung 3.2 zeigt eine Gegenüberstellung des ersten Notensystems von Abbildung 3.1, des zugehörigen Histogramms und der daraus resultierenden horizontalen Segmentierung. Die einzelnen Elemente werden erkannt und separiert. Allerdings werden Notengruppen, die mit einem Balken verbunden sind, nicht weiter segmentiert und können auf Grund der großen Anzahl an möglichen Kombinationen nicht einfach als Bestandteil eines erweiterten Alphabets von dem neuronalen Netz erkannt werden. Wenn eine solche Gruppierung von Noten vorliegt, erkennt das neuronale Netz dieses. Die Gruppe kann dann in einem weiteren Schritt segmentiert werden. Diese Segmentierung erfordert allerdings eine andere Methode.

Probleme bei der Segmentierung

Bereits bei der Verwendung eines Notenblattes in guter Qualität treten bei der Segmentierung Probleme auf.

Eine schiefe, schwache oder verschmutzte Abbildung liefert unter Umständen keine eindeutigen Histogramme und die Notensysteme überschneiden sich. In manchen Notensätzen sind die Systeme vertikal zu dicht gesetzt, so daß sich die Hälse der

Noten zweier übereinander liegender Linien in der Höhe überschneiden (Abbildung 3.3) und so eine einfache Trennung nicht möglich ist. Wenn dieses Problem auftritt kann man das Notenblatt in eine rechte und eine linke Hälfte aufteilen und diese getrennt segmentieren. Dies kann man rekursiv fortsetzen, wenn sich die halben Notensysteme immer noch überschneiden. Hierbei treten zwei Probleme auf: Erstens muß das Programm erkennen, daß Linien nicht segmentiert wurden, und zweitens muß die Rekursion erkennen, wann die Segmentierung beendet ist.

3.2 Die wesentlichen Schritte im Detail

3.2.1 Schätzung der Lage der Notensysteme

Das gescannte Musikstück wird geladen und in eine Schwarzweißgrafik umgewandelt. Danach müssen die einzelnen Notensysteme segmentiert werden. Hierzu wird zunächst die ungefähre Position der Linien geschätzt.

Die Zahl der schwarzen Pixel in jeder Linie wird ermittelt. Dann werden alle Linien auf Null gesetzt, deren Pixelzahl kleiner als das 0,4-fache des Maximums sind. Der Wert 0.4 stellte sich während der Programmierung als praktikabler Kompromiß heraus. Hierdurch wird im Idealfall alles Rauschen durch Notensymbole und insbesondere alle Überschneidungen entfernt und es bleiben nur die Notenlinien über. Zum Segmentieren ist dies nicht geeignet, da alle Symbolteile, die über die Notensysteme hinausragen, ebenfalls entfernt werden.

Der Median des Abstandes der jetzt vorliegenden Spitzen im Histogramm kann als ungefährer Abstand zwischen den einzelnen Notenlinien angenommen werden (da je Notensystem fünf Linien erwartet werden ist der Linienabstand der am häufigsten vorkommende Wert). Wenn man nun einen Tiefpassfilter mit dem sechsfachen Abstand (mit dem sechsfachen Abstand wird das gesamte Notensystem überdeckt) über das Histogramm laufen läßt, erhält man ein neues Histogramm, welches statt einzelner Linien eine Kuppe je Liniensystem aufweist. Zuletzt werden alle Werte ungleich Null auf Eins gesetzt, damit die MATLAB-interne Funktion „findpeaks“ diese Kuppen und ihre Breiten findet. Aus diesen Werten läßt sich Lage der Notensysteme schätzen. Für das Stück in Abbildung 3.3 ist dieser Vorgang graphisch in Abbildung 3.4 dargestellt.

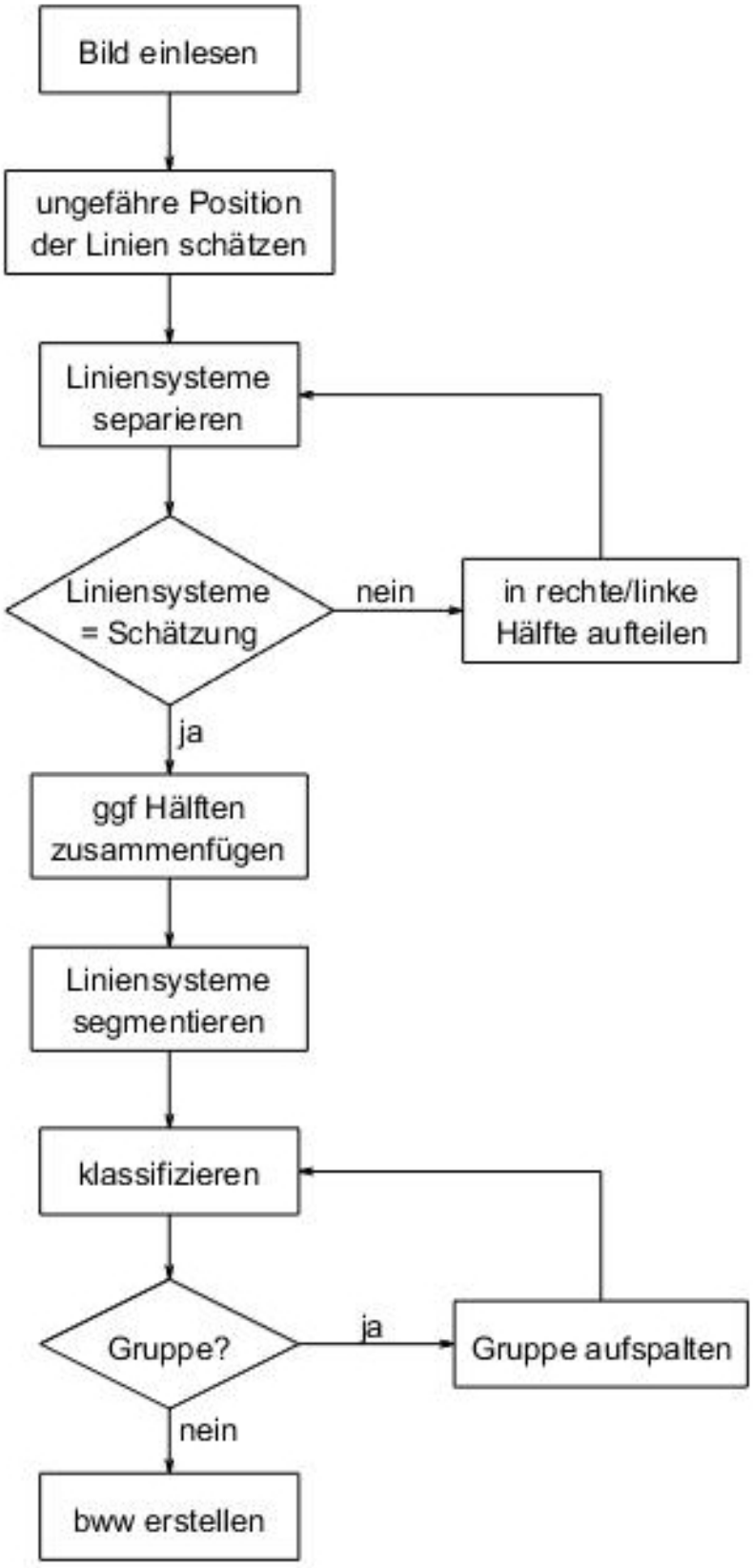


Abbildung 3.5: Ablaufplan der OMR

3.2.2 Notensysteme separieren

Nachdem nun bekannt ist, wo ungefähr die Notensysteme liegen, kann im nächsten Schritt die Segmentierung stattfinden. Hierzu wird wieder die Summe der schwarzen Pixel in jeder Linie betrachtet. Der Algorithmus geht dabei davon aus, daß immer wenn ein Grenzwert über- oder unterschritten wird, ein Notensystem beginnt bzw. endet. Als Grenzwert wurde in diesem Fall drei gewählt. Dementsprechend werden alle Zeilen mit Pixelzahl kleiner als der Grenzwert auf Null und alle anderen Zeilen auf Eins gesetzt. Der Tiefpassfilter entfällt. So kann wieder findpeaks den Anfang und die Höhe der Linie berechnen. Bereiche, deren Höhe kleiner als 24 Pixel ist, können verworfen werden, da in solchen Bereichen Notensysteme nicht sinnvoll dargestellt werden können. Danach wird verglichen, ob in den so berechneten Bereichen ein Notensystem erwartet wurde. Wenn genau eines erwartet wurde, geht der Algoritmus davon aus, daß dieses korrekt separiert wurde. Es kommt aber vor, daß zwei oder mehr erwartet wurden. Dies ist zum Beispiel bei dem Stück in Abbildung 3.3 der Fall, in dem im ersten Schritt der gesamte Notenbereich als ein Segment erkannt wird. Wenn dies passiert, wird der Algorithmus mit der rechten und der linken Hälfte des Bereiches erneut aufgerufen. Wenn nun die erwarteten Notensysteme gefunden wurden, erfolgt die Rückgabe, ansonsten wird die Rekursion eine weiteres Mal durchgeführt. Nach der Rückgabe werden die erhaltenen Bereiche wieder zusammengefügt.

3.2.3 Notensysteme in einzelne Elemente segmentieren

Die Segmentierung der so erhaltenen Notensysteme werden nun nach dem gleichen Verfahren in Einzelelemente segmentiert, wobei in diesem Fall die Summe der schwarzen Pixel in jeder Spalte berechnet wird. Der Grenzwert kann hier nicht auf Null gesetzt werden, da ja die Notenlinien schon für mindestens 5 schwarze Punkte verantwortlich sind. Es zeigte sich, daß das 0,6-fache des Medians des Histogramms hier gute Ergebnisse liefert. Hierbei muß noch beachtet werden, daß dieser Wert nicht unter dem Minimum des Histogramms liegt. In dem Fall muß man den Grenzwert auf diesen Wert erhöhen.

In der Praxis wurden bei schlechten Scans hier auch leere Abschnitte des Notensystems separiert, in denen aber durch Verunreinigungen mehr schwarze Pixel lagen. Diese konnten durch das neuronale Netz als solche erkannt und ignoriert werden.

3.2.4 Klassifikation

Als nächstes werden die so erhaltenen Elemente klassifiziert. Dies kann entweder von Hand (z.B. zum Erzeugen von Trainingsdaten) oder durch das neuronale Netz erfolgen. Hierbei werden jedem Element vier Attribute zugewiesen:

1. Beginn oder Ende einer Überbindung oder Voltenklammer
2. Name des Symbols oder Notenwert
3. Länge der Note (bei Elementen, die keine Länge haben, 0)
4. Ausrichtung des Balkens (auch hier 0, wenn nicht anwendbar)

Wenn eine Notengruppe erkannt wird, muß diese weiter segmentiert werden.

3.2.5 Segmentierung von Gruppen

Eine Notengruppe besteht aus mehreren Symbolen, die vom ersten Algorithmus nicht sauber getrennt werden konnten. Dies kann verschiedene Gründe haben:

- Die Elemente sind so dicht beieinander, daß sie sich überschneiden und daher an keiner Stelle eine Leerstelle (in diesem Falle also nur Notenlinien) entsteht.
- Noten, die durch Balken verbunden sind, werden in den meisten Fällen nicht segmentiert.
- Überbindungen und andere Elemente, die über mehrere Symbole reichen, fallen in diese Kategorie.

Eine weitere Segmentierung mit dem bisherigen Algorithmus mit einem höheren Grenzwert zerschneidet dabei auch einzelne Symbole und führt daher nicht zum Erfolg. Hier hilft der klassische Ansatz, in dem man zunächst die Notenlinien entfernt und dann zusammenhängende Komponenten separiert.

3.2.6 Entfernen der Linien

Zunächst berechnet auch dieser Algorithmus wieder die Anzahl der Punkte in jeder Zeile und setzt alles unterhalb eines Grenzwertes auf Null. Im Idealfall erhält man jetzt die Notenlinien. Probleme treten immer dann auf, wenn die Abbildung von schlechter Qualität ist. Außerdem können Balken und Überbindungen dazu führen, daß zu viele Linien erkannt werden. In diesen Fällen läßt sich das Problem mit heuristischen Methoden lösen.

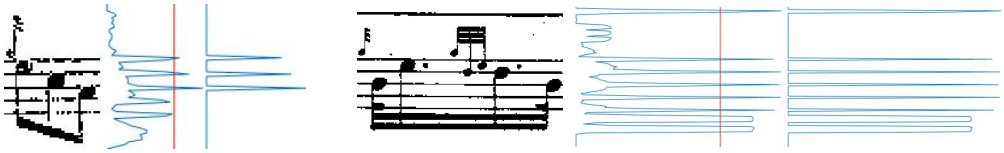


Abbildung 3.6: Graphische Darstellung zum Finden von Linien in Gruppen

Für dieses Problem muß der Grenzwert individuell ermittelt werden. Zunächst wird er auf den Startwert 0.7 gesetzt, der bei guten Bilddaten gut funktioniert. Abbildung 3.6 zeigt zwei Beispiele für Gruppen. Bei der linken werden im ersten Schritt drei Linien erkannt. Zunächst kann man nun den Abstand der Linien ermitteln. Dieser ist meist der Median der Abstände. Wenn aber die weiteren Abstände ungefähr das Zwei- oder Dreifache des kleinsten Abstandes sind, kann dies bedeuten, daß zum Beispiel Linie 1, 2 und 5 entdeckt wurden. In diesem Fall ist der Abstand dann der kleinste errechnete und die Lücken können aufgefüllt werden. Wenn zu viele Linienkandidaten gefunden wurden (wie zum Beispiel im rechten Teil von Abbildung 3.6), müssen diese auf fünf reduziert werden. Hierzu wird zunächst wieder der Median der Abstände ermittelt und alle Kandidaten, die nicht ungefähr in diesem Abstand vom Nachbarn liegen, können eliminiert werden. Als nächstes können Linien, die unterhalb oder oberhalb liegen und deutlich breiter als der Median der Linienbreite sind, entfernt werden, bis fünf Kandidaten übrig bleiben.

Es kann sein, daß man durch dieses Vorgehen keine fünf Notenlinien erhält. In diesem Fall wird der Grenzwert um 0,1 gesenkt und der Vorgang wiederholt. Dies wird solange wiederholt, bis entweder 5 Linien gefunden wurden, oder der Grenzwert unter Null sinkt. Im letzteren Fall wird eine Fehlermeldung ausgegeben und das Element wird ignoriert.

Nachdem die Lage der Linien ermittelt wurde, können diese aus dem Bild entfernt werden. Dies geschieht, indem man jede Pixelspalte betrachtet und die Pixel auf der Höhe der ermittelten Line auf weiß setzt, wenn die benachbarten Pixel schon auf weiß stehen. Wenn diese schwarz sind, löscht man die Linie nicht, da man sonst andere Symbole (z.B. Notenköpfe) zerschneiden würde.

3.2.7 Trennen von Pixelgruppen

Jetzt kann man nach Pixelgruppen suchen, die zusammenhängen oder maximal zwei Pixel weit voneinander entfernt sind, und diese in neuen Grafiken speichern. Die Entfernung von 2 Pixeln (bei Notensystemen, die höher als 75 Pixel sind, wurden 3 Pixel gewählt) werden übersprungen, da bei schlechten Bilddateien zusammenhängende

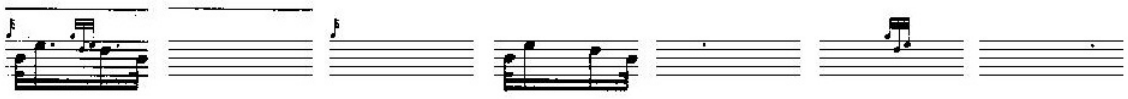


Abbildung 3.7: Gruppe aufgesplittet in zusammenhängende Gruppen

Objekte durch Abbildungsfehler getrennt werden können. Andererseits liegen in den seltensten Fällen Symbole so dicht beieinander, daß sie hier nicht getrennt werden können.

Nach diesem Schritt werden die Notenlinien wieder hinzugefügt.

Abbildung 3.7 zeigt eine solche Segmentierung. Die einzelnen Gruppen zeigen von links nach rechts: Ausgangsgrafik, Teil einer Voltenklammer, einzelne Grace Note, mit Balken verbundene Notengruppe, Punktierung, Doubling, Punktierung. Segmente 2, 4, 5, 6 können auf den eigentlichen Inhalt zurechtgeschnitten und dann klassifiziert werden. Das Segment mit der Notengruppe muß noch einmal weiter aufgespalten werden. Dies kann wieder mit dem bereits mehrfach verwendeten Algorithmus und einem geeignet gewählten Grenzwert erreicht werden. Wie bereits beim Entfernen der Linien muß der Grenzwert hier flexibel gewählt werden, da Notengruppen, abhängig zum Beispiel von der Anzahl der Balken, sehr unterschiedlich viele schwarze Pixel haben können. Hier erwies sich ein Grenzwert vom 0,7-fachen des Maximums des Histogramms als praktikabel. Dieser wurde gegebenenfalls (wenn nicht mindestens 2 Teilelemente gefunden wurden) um 0,1 verringert, um den Prozeß dann erneut durchzuführen.

Bei Elementen, die nicht am Anfang der Gruppe liegen, wird ein paar Pixel vor dem eigentlichen Element geschnitten, da gegebenenfalls nach links gerichtete Balken abgeschnitten würden, wenn direkt am Notenhals getrennt würde.

Bei der Segmentierung von Gruppen nach diesem Verfahren kann es durchaus vorkommen, daß zunächst nur in kleinere Teilgruppen aufgespalten wird, für die dann ein neuer Grenzwert gewählt werden muß. Der Algorithmus versucht hierbei, die lineare Abfolge der einzelnen Elemente von links nach rechts zu erhalten.

Die so erhaltenen Teilsegmente können nun erneut klassifiziert und gegebenenfalls weiter aufgeteilt werden. Danach müssen sie zu einer bww-Datei zusammengesetzt werden. Dieses Zusammensetzen erfolgt mit einer Vielzahl kleiner Heuristiken. Hier ein paar Beispiele:

- Wenn ein neues Notensystem anfängt, muß als erstes ein Notenschlüssel erscheinen. Dieser wird gegebenenfalls hinzugefügt.
- Wenn ein Notensystem endet, muß ein Zeilenende-Symbol kommen.

- Ein Doubling erhält die Notenhöhe der nachfolgenden Note.
- Eine Punktierung erhält die Notenhöhe der vorhergehenden Note.

Kapitel 4

Bewertung des Algorithmus

Zwei Fragestellungen wurden genauer untersucht.

1. Die hier vorgestellte OMR weicht vom Standardvorgehen für OMR im wesentlichen dadurch ab, daß die Separierung der Einzelsymbole in der Regel ohne Entfernen von Notenlinien auskommt. Hier stellt sich die Frage, ob dies einen Einfluß auf die Erkennungsrate hat. Um dies zu testen, wurde das neuronale Netz einmal mit den Symbolen einschließlich der Notenlinien trainiert, und einmal wurden bei jedem Symbol die Notenlinien vor dem Training entfernt. Die Position der Notenlinien wurde in diesem Fall als zusätzliche Daten angehängt. Bei den zu klassifizierenden Bildern wurde entsprechend verfahren.

2. Es stellte sich die Frage, ob das Hinzufügen von künstlichen Trainingsdaten die Erkennungsrate beeinflußt. Hierfür wurden alle vorkommenden Symbole mit Hilfe des Bagpipe Player in verschiedenen Auflösungen ausgedruckt und zur Trainingsmenge hinzugefügt. Getestet wurden dann sowohl Netze, die mit künstlich erzeugten Daten trainiert wurden, als auch solche, bei denen dies nicht der Fall war.

Hierbei wurde allerdings auf die Wiederholung der Versuche verzichtet. Um genauere, statistisch sicherere Resultate zu erhalten, müßten diese Tests mehrfach wiederholt werden. Dies war im Rahmen dieser Arbeit nicht möglich.

4.1 Netzgröße

Im Rahmen dieser Arbeit wurde nicht versucht, die Größe des neuronalen Netzes auf das vorliegende Problem hin zu optimieren. Die Zielsetzung war an dieser Stelle nur, ein Netz zu finden, welches die gegebenen Symbole klassifizieren kann.

Zum Trainieren des Netzes wurde die in MATLAB implementierte Funktion „train“ verwendet. Mit den hier verwendeten Standardparametern teilt train die Trainingsmenge zufällig in drei Teilmengen auf. Die größte, die 70% der Trainingsmenge enthält, wird zum Trainieren des Netzes mit Hilfe der Backpropagation verwendet. Der Backpropagation Algorithmus findet ein (möglicherweise lokales) Fehlerminimum für die Trainingsmenge. Hierbei ist aber das Problem, daß dies zu Überanpassungen des Netzes an die Trainingsmenge führen kann und die Erkennungsrate auf neuen Eingaben unter Umständen schlechter ist. Zu diesem Zweck wird die zweite Teilmenge (15%) in jedem Schritt ausgewertet. Die Backpropagation stoppt, wenn dieser Wert nicht mehr besser wird. Die letzten 15% werden zur Bewertung des erhaltenen Netzes genommen. Als Eingabe für das neuronale Netz wurden die Bilder der Einzelsymbole auf eine Standardgröße von 60 mal 20 Bildpunkten umgerechnet.

Es stellte sich durch Sichtprüfung heraus, daß ein Netz mit 70 Neuronen in der ersten Schicht kein brauchbares Ergebnis erzeugte. Die bww-Stücke wurden mit 150 Neuronen in der ersten Schicht wiedererkennbar umgewandelt. Für die Stücke aus anderen Quellen waren die Ergebnisse hier nicht lesbar und wurden bei der Auswertung nicht berücksichtigt. Netze mit zwei versteckten Schichten mit je 250 Neuronen lieferten Ergebnisse, die zumindest mit der Metrik bewertet werden konnten. Größere Netze wurden testweise trainiert, lieferten aber bei Sichtprüfung keine besseren Ergebnisse und wurden daher nicht weiter ausgewertet.

Alle weiteren Experimente wurden mit neuronalen Netzen durchgeführt, die zwei versteckte Schichten mit je 250 Knoten enthielten. Eine weitere Optimierung der Netzgröße wurde hier nicht vorgenommen.

4.2 Versuchsrahmen

Um eine OMR-Software zu bewerten, gibt es kein etabliertes Standardverfahren [3].

In der vorliegenden Arbeit wird die von Bellini et al. in ihrem Artikel [3] vorgeschlagene Methode angewendet. Die Autoren vergleichen dabei das zu erkennende Notenblatt mit einem von der OMR-Software erzeugten Bild der Noten, indem sie jedes Element der Notenschrift abgleichen. Der Abgleich wird dabei von Hand vorgenommen. Hierbei werden alle einzelnen Symbole gegebenenfalls in Teilsymbole aufgeteilt. Eine Note besteht dann aus dem Notenkopf, der die Tonhöhe angibt, dem Notenhals und den Fähnchen/Balken, die die Länge und bei Balken noch die Ausrichtung bestimmen. Notenschlüssel, Taktstriche oder Vorzeichen werden nicht weiter aufgeteilt. Nachdem die Einzelelemente abgeglichen worden sind, werden die

richtig und die falsch erkannten Zeichen jeweils gezählt. Eine genauere Beschreibung der angewandten Metrik erfolgt in Abschnitt 4.3.

Um das Netz zu trainieren wurden 15 Musikstücke aus unterschiedlichen Quellen gescannt und von Hand klassifiziert. Außerdem wurden aus dem Bagpipe Player heraus alle vorkommenden Symbole in verschiedenen Auflösungen ausgedruckt und klassifiziert. Hieraus resultierten knapp 15000 Trainingssegmente für die neuronalen Netze, von denen ca. 6400 aus bww-Dateien erzeugt wurden.

Für die Versuche wurden insgesamt sechs Musikstücke mit unterschiedlichem Druckbild ausgewählt. Zwei Stücke stammen aus den „Scots Guards I“ [1]. Dies ist eine Notensammlung, die eine große Zahl an musikalischen Standards für den Dudelsack enthält. Die gleichen Stücke wurden auch aus dem Bagpipe Player ausgedruckt und sowohl als Bilddatei als auch als Scan ausgewertet. Weiterhin wurde eine Stück aus „The Moidart Collection“ [9] und eines aus dem Buch „A Bibliography for Bagpipe Music“ [6], zu dem nur eine Bilddatei vorlag (siehe Abbildung 3.3), verwendet. Letzteres war in der Bildqualität deutlich schlechter als die anderen Scans.

4.3 Metrik

Zur Bewertung der Ergebnisse wurde die schon erwähnte, aus dem Artikel von Bellini et al. stammende Methode verwendet. Wie die Autoren in ihrer Einleitung feststellen, gibt es zur Zeit kein etabliertes Standardverfahren zur Bewertung von Notenerkennungsprogrammen. Sie identifizieren hierbei folgende Probleme:

1. Notation von Musik ist nicht eindeutig definiert. Manche Dinge können bei gleicher Bedeutung auf unterschiedliche Art notiert werden. Ist hierbei die korrekte Sinnerkennung oder die symbolgenaue Wiedergabe zu bewerten?

2. Innerhalb einer Computerrepräsentation für Noten können optisch identische Notengebilde sehr unterschiedlich dargestellt werden.

3. Es gibt kein Standardformat, um Musik auf Computern zu repräsentieren. Jeder Hersteller von OMR-Programmen benutzt sein eigenes Format.

4. Die vorhandenen Programme haben teils sehr unterschiedliche Zielsetzungen, die schwer vergleichbar sind. Wie soll man ein Programm zur Erzeugung von Midi-Dateien mit einem Notensatzprogramm vergleichen?

5. Als wie schwerwiegend sind unterschiedliche Erkennungsfehler zu bewerten? Ist die Erkennung der Notenlänge wichtiger als die Erkennung, ob die Note in einer Gruppe notiert ist?

6. Viele Hersteller von OMR-Programmen schreiben ihrer Software eine Erkennungsrate von 90% und mehr zu. Oft ist aber nicht klar, welche Bewertungsmethode angewandt und welche Notenvorlagen verwendet wurden. Wie soll man die Ergebnisse vergleichen?

7. Es gibt keinen Kanon von Musikstücken, der zur Standardisierung der Bewertung herangezogen werden kann.

Die Autoren schlagen daher vor, nicht die Computerrepräsentation zu vergleichen, sondern die aus dem Scan erzeugten Daten wieder auszudrucken und das so erhaltene Resultat mit dem ursprünglichen Notenbild zu vergleichen. Sie schlagen hierzu zwei Metriken vor.

In der ersten Metrik sollen elementare Symbole wie Notenhöhe, Notenlänge, Vorsetzungszeichen etc. miteinander verglichen werden.

Hierbei kommen folgende Werte zur Berechnung (die Bezeichnungen sind hierbei im wesentlichen von Bellini übernommen):

$NEBS$: Zahl der erwarteten elementare Symbole (Number of expected basic symbols).

$NTBS$: Zahl der auf dem OMR-Blatt korrekten dargestellten elementaren Symbole (Number of true basic symbols).

$NFBS$ Zahl der auf dem OMR-Blatt falsch dargestellten elementaren Symbole (Number of falsely recognized basic symbols). Symbole, die fehlerhaft erkannt wurden, wie zum Beispiel falsch Notenhöhe oder -länge.

$NABS$ Zahl der auf dem OMR-Blatt hinzugefügten elementaren Symbole (Number of added basic symbols). Hier werden Symbole gezählt, die auf dem ursprünglichen Notenblatt nicht vorkommen.

$NMBS$ Zahl der auf dem OMR-Blatt fehlenden elementaren Symbole (Number of missed basic symbols).

$NBSE = NFBS + NMBS + NABS$ Summe der Fehler („number of basic symbol errors“)

Die Autoren definieren hier $NTBS$ als die korrekt positiven und $NFBS + NMBS$ als die falsch negativen Symbole, während $NABS$ die falsch positiven Symbole repräsentieren. Korrekt negative Symbole sind in diesem Zusammenhang nicht

sinnvoll, da alle auf einem Notenblatt vorhandenen Symbole als relevant betrachtet werden.

Da die Software allerdings alle ausgegebenen Symbole als korrekt ansieht, erscheint es sinnvoller, die Summe aller nicht korrekt ausgegebenen Symbole, also $NFBS + NABS$ als falsch positiv zu definieren und die ausgelassenen Symbole $NMBS$ als falsch negativ zu betrachten. Hier werden im Weiteren letztere Definitionen verwendet.

Nun kann man Angaben zur Rate der korrekt und falsch erkannten sowie der hinzugefügten Symbole wie folgt berechnen:

Anteil der korrekt erkannten Symbole (true basic symbol rate):

$$TBSR = \frac{NTBS}{NEBS}$$

Anteil der falsch erkannten Symbole (false basic symbol rate):

$$FBSR = \frac{NFBS}{NEBS}$$

Anteil der nicht erkannten Symbole (missed basic symbol rate):

$$MBSR = \frac{NMBS}{NEBS}$$

Anteil der hinzugefügten Symbole (added basic symbol rate):

$$ABSR = \frac{NABS}{NEBS}$$

Die Gesamtfehlerrate ist dann (basic symbol error rate):

$$BSER = FBSR + MBSR + ABSR$$

Desweiteren können die in der Forschung zum maschinellen Lernen gebräuchlichen Begriffe „Precision“ und „Recall“ nun wie üblich definiert werden, wobei hier auf die alternative Definition für korrekt positiv, falsch positiv und falsch negativ zurückgegriffen wird. Außerdem erscheint es sinnvoller, den Recall wie in Formel 4.2 zu definieren. Recall gibt hierbei an, wie viele der erkennbaren Symbole korrekt erkannt wurden, während Precision angibt, wie viele der erkannten Symbole korrekt waren.

$$Precision = \frac{korrekt\ positive}{korrekt\ positiv + falsch\ positiv} = \frac{NTBS}{NTBS + NFBS + NABS}$$

(4.1)

$$Recall = \frac{korrekt\ positve}{erwartete\ Symbole} = \frac{NTBS}{NEBS} = TBSR$$

(4.2)

Für eine genauere Analyse kann man die Fehler noch für einzelne Symbolklassen separiert betrachten und so feststellen, welche Elemente wie gut erkannt werden. Diese gesplitteten Ergebnisse kann man dann noch unterschiedlich gewichten. Hierzu wurden von den Autoren Expertenmeinungen eingeholt und verschiedene Gewichte vorgeschlagen. Auf eine Wichtung wird im Rahmen dieser Arbeit verzichtet, da zum gegebenen Zeitpunkt kein zusätzlicher Informationsgewinn durch die Wichtung zu erwarten ist.

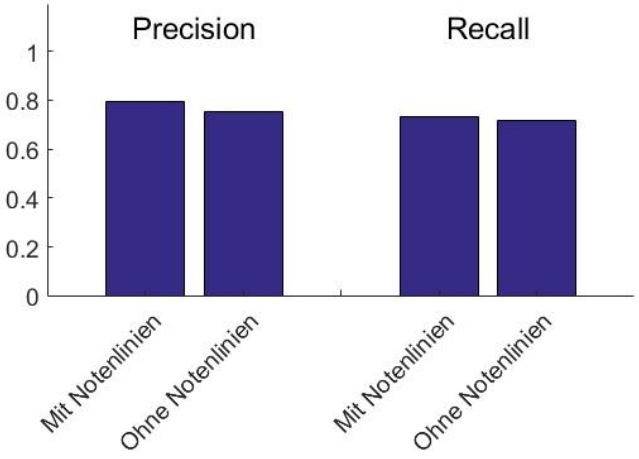
In der zweiten von Bellini vorgeschlagenen Metrik werden nicht die elementaren Symbole, sondern die daraus zusammengesetzten Symbole betrachtet. In dieser Metrik werden dann nur vollständig korrekt erkannte Noten inklusive Höhe, Länge, Zugehörigkeit zu einer Gruppe, Ausrichtung des Balkens etc. als korrekt gewertet. Die für diese Metrik berechneten Werte werden analog zur elementaren Metrik bezeichnet, wobei allerdings das „B“ für basic durch ein „C“ für composite verwendet wird. Im Rahmen dieser Arbeit wird auf diese Metrik verzichtet, da es zum einen keine spezialisierte Software gibt, mit der ein direkter Vergleich möglich wäre, und zum anderen an dieser Stelle keine weitere Erkenntnis hiervon erwartet wird.

4.4 Interpretation der Ergebnisse

Im Folgenden soll nun die Metrik für elementare Symbole ausgewertet werden. Hierzu wurden, wie bereits in Abschnitt 4.2 beschrieben, 6 Stücke ausgewählt und durch verschiedene neuronale Netze umgewandelt. Tabellen mit den detaillierten Ergebnissen, aufgelistet nach verschiedenen Kategorien, finden sich im Anhang A.

4.4.1 Notenlinien

Als erstes wurde untersucht, ob das Entfernen der Notenlinien zu einer Verbesserung der Klassifizierung führt. Hierfür wurden zwei neuronale Netze trainiert. Für



	Precision	Recall
Mit	0,7939	0,7328
Ohne	0,7516	0,7177

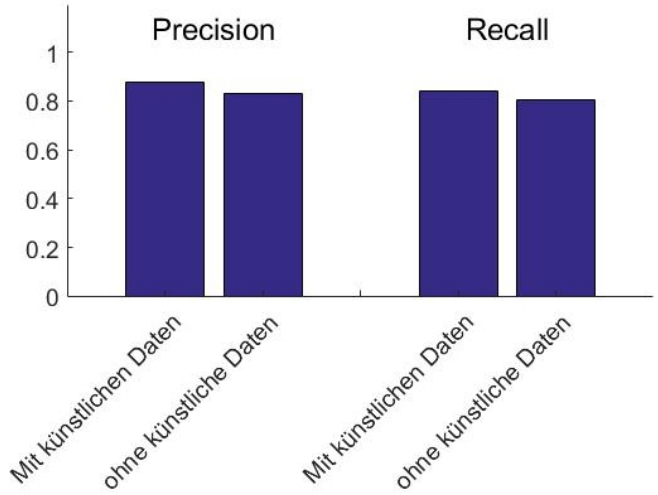
Abbildung 4.1: Vergleich der Klassifizierung mit und ohne Notenlinien

das erste wurden die Bilder der Segmente direkt gelernt und klassifiziert. Für das zweite wurden zunächst in jedem Segment mit dem in Kapitel 3.2.5 beschriebenen Algorithmus die Notenlinien entfernt. Mit den so erhaltenen Bildern, denen die Position der Notenlinien als weitere Werte angehängt wurden, wurde dann das neuronale Netz trainiert.

Die aus diesen Experimenten gewonnen Notenbilder wurden nun bewertet. Abbildung 4.1 zeigt die erhaltenen Werte für Precision und Recall für dieses Experiment. Es zeigt sich, daß das Entfernen der Notenlinien sogar eine Verschlechterung der Ergebnisse herbeiführt.

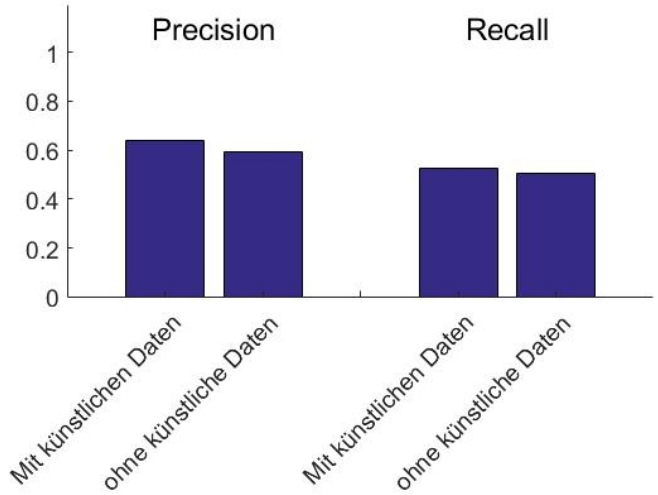
4.4.2 Künstliche Trainingsdaten

Als nächstes wurde untersucht, ob das Hinzufügen von künstlich erzeugten Daten einen Vorteil bringt. Hierzu wurde das neuronale Netz einmal mit allen verfügbaren Daten und einmal nur mit den aus Musikstücken erhaltenen trainiert. Die in verschiedenen Auflösungen ausgedruckten Beispiele für alle verfügbaren Symbole wurden im zweiten Fall weggelassen. Abbildung 4.2 und 4.3 zeigen die Ergebnisse getrennt für die Stücke aus dem Bagpipe Player und die Stücke aus anderen Quellen. Hier zeigt sich, daß mehr Trainingsdaten in beiden Kategorien bessere Ergebnisse liefern.



	Precsion	Recall
Mit	0,8756	0,8402
Ohne	0,8312	0,8030

Abbildung 4.2: Vergleich der Resultate für Netze mit und ohne künstliche Daten bei Stücken aus dem Bagpipe Player



	Precsion	Recall
Mit	0,6381	0,5238
Ohne	0,5916	0,5059

Abbildung 4.3: Vergleich der Resultate für Netze mit und ohne künstliche Daten bei den Scans aus Notenbüchern

4.4.3 Fehleranalyse

Zuletzt kann man noch betrachten, welche Fehler auftraten. Hierzu ist es hilfreich, die Fehler nach Kategorie aufzuschlüsseln. Detaillierte Zahlen hierzu finden sich im Anhang A.

Abbildung 4.4 zeigt, wie hoch der prozentuale Fehler innerhalb jeder Kategorie ist. Die Kategorien, die am schlechtesten erkannt wurden, sind die der Taktbezeichner und der sonstigen Verzierungen. Dies ist allerdings nicht verwunderlich, wenn man die Zahl an Trainingsbeispielen in Abbildung 4.5 betrachtet. Die beiden genannten Kategorien sind die, für die nur sehr wenige Trainingsbeispiel vorlagen. Taktbezeichner kommen üblicherweise nur einmal in jedem Musikstück vor, und sonstige Verzierungen bezeichnet eine große Zahl an weniger üblichen Verzierungen.

Betrachtet man die Fehlerraten genauer, so fällt auf, daß viele Fehler bei der Ermittlung der Notenhöhe (42%) und der Notenlänge (34%) gemacht werden. Gerade in diesen Kategorien ist die Zahl der Trainingsbeispiele aber sehr hoch (je 26% aller Trainingsbeispiele). Andererseits werden Taktstriche (14% Fehler) und Doublings (11% Fehler) recht gut erkannt, obwohl sie nur ca. 7% respektive 3% der Trainingsmenge ausmachen. Während es sich bei Taktstrichen um genau 1 Symbol handelt, umfaßt die Kategorie Doublings 16 unterschiedliche Symbole, die aber eine sehr einheitliche Struktur aufweisen und sich im wesentlichen durch die Höhe einer Note unterscheiden. (Abbildung 4.9 zeigt die verfügbaren Doublings). Dies deutet an, daß bei der Erkennung der Notenhöhe und -länge andere Probleme vorliegen. Abbildung 4.8 zeigt, daß der Erkennungsfehler hier deutlich unterschiedlich ist, wenn man den anteiligen Fehler zwischen Stücken aus dem Bagpipe Player und solchen aus anderen Quellen vergleicht. Abbildung 4.10 zeigt den Vergleich eines Taktes einmal als Ausdruck aus dem Bagpipe Player (links) und einmal aus den Scots Guards (rechts). Man sieht deutlich, daß bei den Noten aus dem Bagpipe Player alle Notenhälse bis auf die gleiche Ebene verlängert und dementsprechend die Balken alle auf gleicher Höhe dargestellt werden. Dies erleichtert es der Erkennungssoftware deutlich, die Notenlänge anhand der Balken zu ermitteln, da die notwendige Information immer im gleichen Teilbereich des Bildes zu suchen ist. Im Gegensatz dazu sind die Notenhälse in den Scots Guards deutlich kürzer, und die Balken haben nicht nur unterschiedliche Höhen, sondern verlaufen auch unterschielich schräg und teilweise durch die Notenlinien. Dies erschwert die Erkennenung der Notenlänge erheblich.

Ein weiterer Effekt entsteht durch die unterschiedlichen Längen der Notenhälse: Die Höhe des segmentierten Notensystems innerhalb der Bildsegments ist nicht klar definiert. Während bei Notationen aus dem Bagpipe Player das untere Ende der

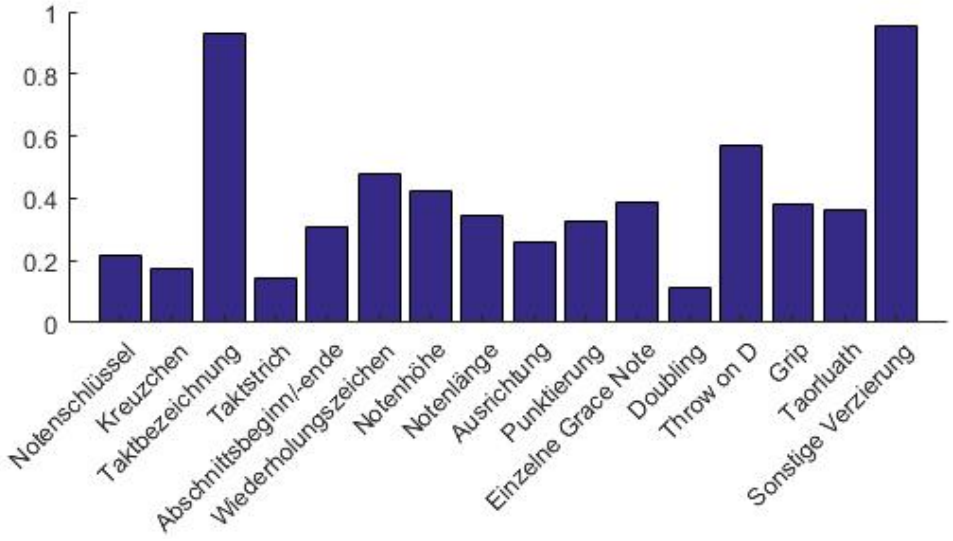


Abbildung 4.4: Anteiliger Fehler innerhalb der Kategorien

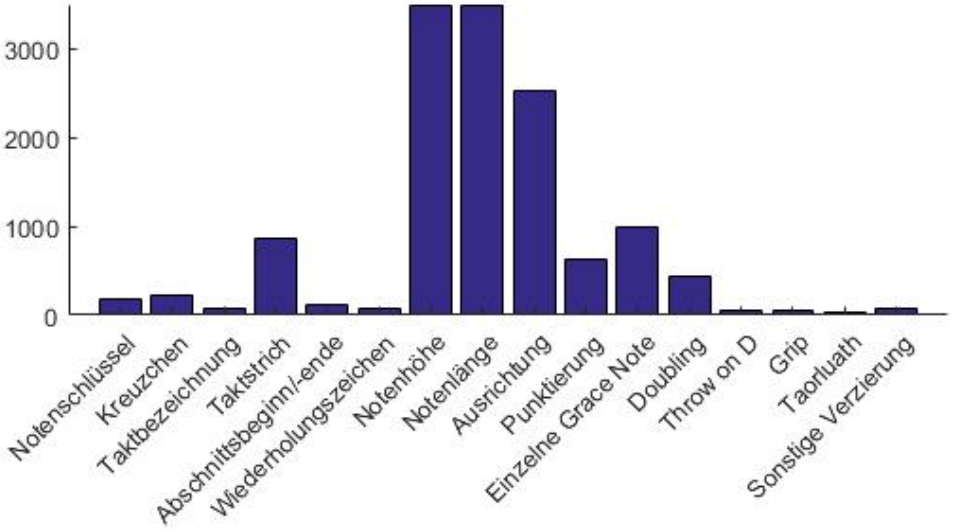


Abbildung 4.5: Anzahl Trainingsbeispiele

Notenzeile eine definierte, immer gleiche Höhe zum Notensystem hat, ist dies bei den anderen Quellen nicht sicher gewährleistet. Daher ist die relative Position der Notenlinien innerhalb des gegebenen Segments nicht standardisiert. Dies führt wiederum dazu, daß die Notenköpfe bei gegebener Notenhöhe keine Standardhöhe im Bild haben und daher deutlich schwerer erkannt werden als dies bei Systemen aus dem Bagpipe Player der Fall ist.

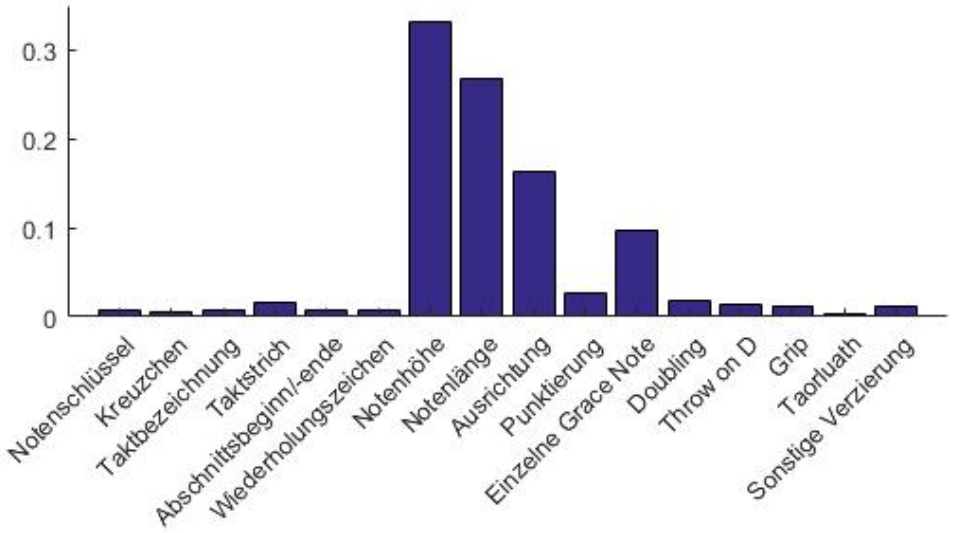


Abbildung 4.6: Anteil der Kategorie am Gesamtfehler

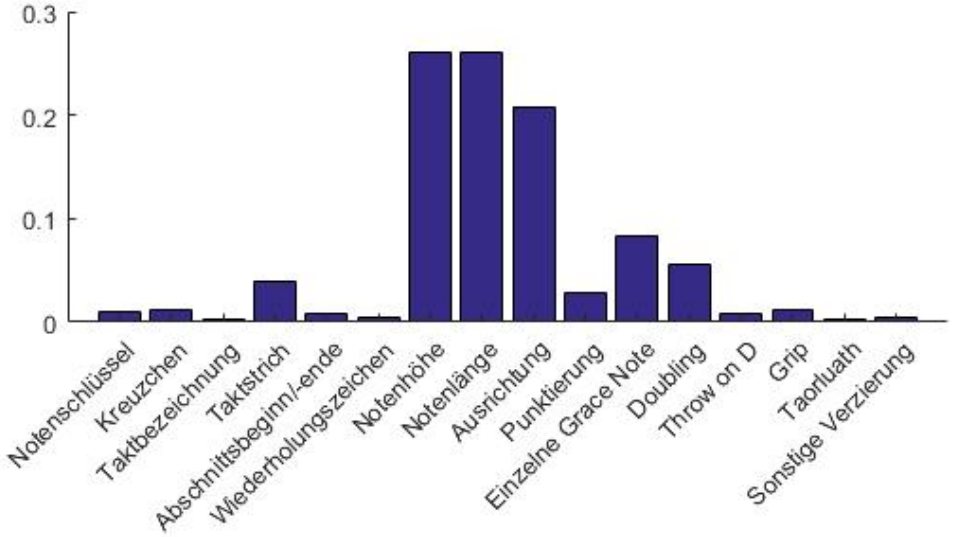
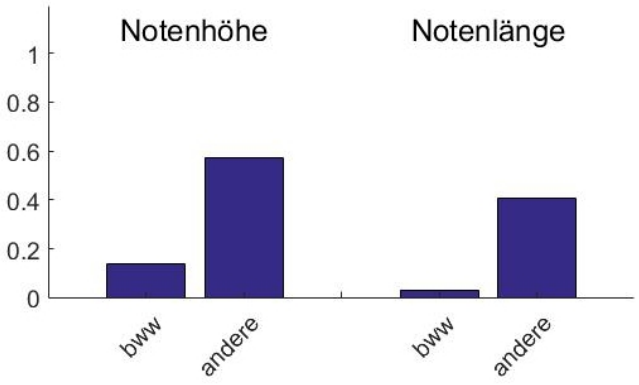


Abbildung 4.7: Anzahl der Symbole in den Teststücken



	bww	andere
Notenhöhe	0,1360	0,5728
Notenlänge	0,0318	0,4043

Abbildung 4.8: Vergleich der Fehlerrate von Notenhöhen und Notenlinien bei bww-Dateien und Scans aus Notenbüchern

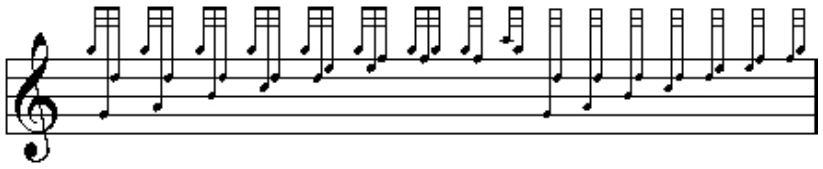


Abbildung 4.9: Doublings

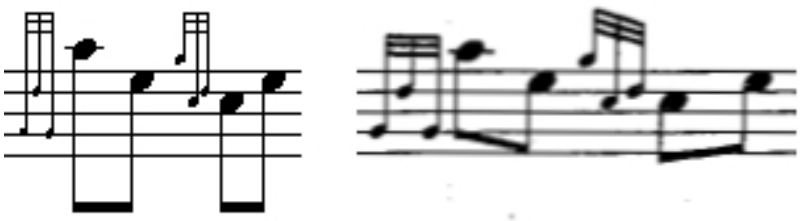


Abbildung 4.10: Vergleich eines Taktes: links bww, rechts Scots Guards

Kapitel 5

Fazit

5.1 Zusammenfassung

Unter Idealbedingungen (direkt erzeugte Bilddatei aus bww) ist die Erkennungsrate (Recall) mit über 90% gut, und auch die Precision liegt über der 90%-Marke. Hier erreichten die Netze mit einer versteckten Schicht mit 150 Neuronen fast so gute Ergebnisse wie die größeren Netze mit 250+250 Neuronen. Für die gescannten Bilddateien waren die Ergebnisse mit den kleineren Netzen allerdings nicht lesbar.

Die Resultate werden deutlich schlechter, wenn man von den Idealbedingungen abweicht. Ein direkter Vergleich der bww-Stücke, einmal als direkte Bilddatei und einmal als Scan, zeigt, daß die Fehlerrate von 5-7% auf über 30% ansteigt.

Lieder, die nicht aus bww-Dateien erzeugt wurden, sondern aus anderen Quellen stammen, werden noch schlechter erkannt und die Fehlerrate steigt in den meisten Fällen über 50%.

Die vorliegenden Daten deuten an, daß das Entfernen der Notenlinien, wie dies im klassischen Algorithmus durchgeführt wird, für die Erkennung nicht notwendig ist, wenn neuronale Netze zur Klassifizierung verwendet werden und die Segmentierung anderweitig erfolgen kann. Auch ein Zerteilen der Symbole in einzelne Primitive (Ovale, Quader etc.) erweist sich als unnötig, da das neuronale Netz diese Teilinformationen selbständig erkennt. Allerdings reichen die vorliegenden Daten nicht aus, hierüber eine endgültige Aussage zu treffen. Bei den Netzen mit 250+250 versteckten Knoten bringt es eine leichte Verbesserung, bei denen mit 150 versteckten Knoten werden die Ergebnisse schlechter. Der Unterschied ist allerdings in beiden Fällen nicht sonderlich hoch. Um hier eine fundierte Aussage treffen zu können, müßten mehr Tests durchgeführt werden.

Insbesondere die schlechte Erkennung der Notenhöhe und -länge bei Stücken, die nicht aus dem Bagpipe Player stammen, führt zu der Annahme, daß hier eine Standardisierung der Höhe der Notenlinien innerhalb des Bildsegments zu einer deutlichen Verbesserung mindestens der Notenhöhe führen könnte. Dies sollte in weiteren Versuchen geklärt werden.

5.2 Ausblick

Die Normierung der Lage der Notenlinien innerhalb der Bildsegmente erscheint als der naheliegende und erste Schritt für eine Verbesserung der Erkennungsergebnisse.

Die hier durchgeführten Tests mußten von Hand ausgewertet werden. Der zeitliche Aufwand hierfür ist erheblich und macht eine größere Studie sehr aufwendig. Da es sich bei bww-Dateien aber um reguläre Ausdrücke handelt, sollte es möglich sein, die Tests zu automatisieren. Durch mehr Tests könnten genauere Aussagen gemacht werden, ob zum Beispiel das Entfernen von Notenlinien tatsächlich keinen Unterschied hinsichtlich der Erkennungsrate macht.

In der Nachbearbeitung der durch das neuronale Netz erhaltenen Symbole können diverse in den Tests aufgetretene Fehler aufgefangen werden. So konnten zum Beispiel einige Notenschlüsseln eingefügt werden, die nicht richtig erkannt wurden, da das Programm „weiß“, daß am Anfang einer Zeile ein Notenschlüssel stehen muß. Diese Einfügung ist für die Darstellung im Bagpipe Player notwendig, da das Programm Symbole, die nicht zwischen Notenschlüssel und dem Symbol für das Zeilenende stehen, schlichtweg ignoriert. In der Nachbearbeitung könnten diverse weitere syntaktische Fehler aufgefangen oder zumindest angezeigt werden.

Neben syntaktischen Fehlern können teilweise auch semantische Fehler aufgespürt werden, indem unsinnige Symbolkombinationen (z.B. zwei Verzierungen direkt hintereinander) oder Takte mit zu wenigen oder zu vielen Noten erkannt werden. Hier können bei auftretenden Widersprüchen auch die Klassifikationen von Elementen genauer betrachtet werden. Der vorliegende Algorithmus wählt die Klassifikation mit der höchsten Konfidenz, bei Widersprüchen könnte aber eruiert werden, ob Symbolkombinationen mit einer etwas geringeren Konfidenz in den einzelnen Klassifikationen die Widersprüche auflösen können.

Die Segmentierung der Notensysteme wurde auf die traditionelle Weise durch Zählen von schwarzen Pixeln erreicht, die dann heuristisch untersucht wurden. Hier könnte versucht werden, auch dies von einem lernenden System durchführen zu

lassen. Auch die Segmentierung der Einzelelemente könnte durch ein solches System erfolgen.

Abschließend läßt sich feststellen, daß der vorgestellte Ansatz grundsätzlich funktioniert, und daß mit den hier aufgeführten Ansätzen noch eine deutliche Verbesserung dieser OMR realisierbar erscheint.

Anhang A

Detaillierte Versuchsergebnisse

Es folgen nun die Ergebnisse der Tests, aufgeschlüsselt nach elementaren Symbolen. Es wurde unterschieden nach den wesentlichen Symbolen zum Aufbau des Notensystems, den Noten aufgeschlüsselt nach ihren Informationsmerkmalen und den wichtigsten Verzierungen.

In den Spalten ist jeweils die Häufigkeit des Auftreten der jeweiligen Ereignisse aufgeführt. Hierbei werden die folgenden Abkürzungen verwendet:

- $NEBS$: Zahl der erwarteten elementaren Symbole
- $NTBS$: Zahl der korrekten elementaren Symbole
- $NFBS$: Zahl der falschen elementaren Symbole
- $NABS$: Zahl der hinzugefügten elementaren Symbole
- $NMBS$: Zahl der nicht erkannten elementaren Symbole
- $NBSE$: Summe der Fehler

- $TBSR$: Anteil der richtig erkannten Symbole
- $FBSR$: Anteil der falsch erkannten Symbole
- $MBSR$: Anteil der nicht erkannten Symbole
- $ABSR$: Anteil der hinzugefügten Symbole
- $BSER$: Gesamtfehler ($FBSR + MBSR + ABSR$)

Green Hills of Tyrol

direkte bww-grafik

150 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	4	0	0	0	0
Notenhöhe	77	68	9	0	0	9
Notenlänge	77	77	0	0	0	0
Ausrichtung	58	58	0	0	0	0
Punktierung	11	11	0	0	0	0
Einzelne Grace Note	25	20	5	0	0	5
Doubling	20	20	0	0	0	0
Throw on D	2	2	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	295	16	0	0	16

TBSR	0,9486
FBSR	0,0514
MBSR	0
ABSR	0
BSER	0,0514

Precision	0,9486
Recall	0,9486

Green Hills of Tyrol

direkte bww-grafik

150 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	2	2	0	0	2
Notenhöhe	77	72	5	0	0	5
Notenlänge	77	69	8	0	0	8
Ausrichtung	58	57	1	0	0	1
Punktierung	11	11	0	0	0	0
Einzelne Grace Note	25	20	5	0	0	5
Doubling	20	19	0	1	0	1
Throw on D	2	2	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	1	0	0	0	0
	311	288	22	1	0	23

TBSR	0,9260
FBSR	0,0707
MBSR	0,0032
ABSR	0
BSER	0,0739

Precision	0,9290
Recall	0,9260

Green Hills of Tyrol

direkte bww-grafik

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	2	2	0	0	2
Wiederholungszeichen	4	2	2	0	0	2
Notenhöhe	77	72	5	0	0	5
Notenlänge	77	70	7	0	0	7
Ausrichtung	58	58	0	0	0	0
Punktierung	11	11	0	0	0	0
Einzelne Grace Note	25	20	5	0	0	5
Doubling	20	20	0	0	0	0
Throw on D	2	2	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	288	23	0	0	23

TBSR	0,9260
FBSR	0,0740
MBSR	0
ABSR	0
BSER	0,0740

Precision	0,9260
Recall	0,9260

Green Hills of Tyrol

direkte bww-grafik

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	4	0	0	0	0
Notenhöhe	77	75	2	0	0	2
Notenlänge	77	75	2	0	0	2
Ausrichtung	58	58	0	0	0	0
Punktierung	11	11	0	0	0	0
Einzelne Grace Note	25	17	8	0	0	8
Doubling	20	20	0	0	0	0
Throw on D	2	0	2	0	0	2
Grip	5	5	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	295	16	0	0	16

TBSR	0,9486
FBSR	0,0514
MBSR	0
ABSR	0
BSER	0,0514

Precision	0,9486
Recall	0,9486

Green Hills of Tyrol

Scan aus bww-Ausdruck

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	2	2	0	0	2
Kreuzchen	8	4	3	1	0	4
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	9	3	2	0	5
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	0	2	2	0	4
Notenhöhe	77	42	22	13	1	36
Notenlänge	77	58	6	13	1	20
Ausrichtung	58	44	2	12	0	14
Punktierung	11	10	0	1	1	2
Einzelne Grace Note	25	15	7	3	2	12
Doubling	20	15	4	1	0	5
Throw on D	2	1	0	1	0	1
Grip	5	3	2	0	0	2
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	207	55	49	5	109

TBSR	0,6656
FBSR	0,1768
MBSR	0,1576
ABSR	0,0161
BSER	0,3505

Precision	0,7753
Recall	0,6656

Green Hills of Tyrol

Scan aus bww-Ausdruck

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	2	2	0	0	2
Kreuzchen	8	5	3	0	0	3
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	9	3	2	0	5
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	3	1	0	0	1
Notenhöhe	77	44	27	6	4	37
Notenlänge	77	64	7	6	4	17
Ausrichtung	58	50	4	4	0	8
Punktierung	11	10	0	1	0	1
Einzelne Grace Note	25	15	10	0	1	11
Doubling	20	18	2	0	0	2
Throw on D	2	1	0	1	0	1
Grip	5	1	4	0	0	4
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	226	65	20	9	94

TBSR	0,7267
FBSR	0,2090
MBSR	0,0643
ABSR	0,0289
BSER	0,3023

Precision	0,7533
Recall	0,7267

Green Hills of Tyrol

Scan aus Scots Guards I

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	1	3	0	1	4
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	9	1	2	0	3
Abschnittsbeginn/-ende	4	2	2	0	0	2
Wiederholungszeichen	4	1	0	3	0	3
Notenhöhe	77	28	20	29	2	51
Notenlänge	77	31	17	29	2	48
Ausrichtung	58	34	2	22	0	24
Punktierung	10	2	0	8	0	8
Einzelne Grace Note	25	9	6	10	0	16
Doubling	20	17	2	1	0	3
Throw on D	3	2	0	1	0	1
Grip	5	2	1	2	0	3
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	303	138	56	107	5	168

TBSR	0,4554
FBSR	0,1848
MBSR	0,3531
ABSR	0,0165
BSER	0,5545

Precision	0,6935
Recall	0,4554

Green Hills of Tyrol

Scan aus Scots Guards I

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	0	1	0	1
Taktstrich	14	13	1	1	0	2
Abschnittsbeginn/-ende	4	1	3	1	0	4
Wiederholungszeichen	4	1	2	1	0	3
Notenhöhe	77	32	31	14	6	51
Notenlänge	77	33	22	14	6	42
Ausrichtung	58	32	16	10	1	27
Punktierung	10	4	1	5	0	6
Einzelne Grace Note	25	10	7	8	0	15
Doubling	20	18	0	2	0	2
Throw on D	3	0	2	1	0	3
Grip	5	1	2	2	0	4
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	303	149	88	60	13	161

TBSR	0,4917
FBSR	0,2904
MBSR	0,1980
ABSR	0,0429
BSER	0,5314

Precision	0,5960
Recall	0,4917

Green Hills of Tyrol

direkte bww-grafik

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	4	0	0	0	0
Notenhöhe	77	62	15	0	0	15
Notenlänge	77	65	12	0	0	12
Ausrichtung	58	58	0	0	0	0
Punktierung	11	11	0	0	0	0
Einzelne Grace Note	25	16	8	1	0	9
Doubling	20	20	0	0	0	0
Throw on D	2	0	2	0	0	2
Grip	5	5	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	271	39	1	0	40

TBSR	0,8714
FBSR	0,1254
MBSR	0,0032
ABSR	0
BSER	0,1286

Precision	0,8742
Recall	0,8714

Green Hills of Tyrol

Scan aus bww-Ausdruck

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	2	2	0	0	2
Kreuzchen	8	4	4	0	0	4
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	10	4	0	0	4
Abschnittsbeginn/-ende	4	4	0	0	0	0
Wiederholungszeichen	4	2	1	1	0	2
Notenhöhe	77	44	26	7	1	34
Notenlänge	77	58	13	6	1	20
Ausrichtung	58	49	1	8	1	10
Punktierung	11	11	0	0	2	2
Einzelne Grace Note	25	15	7	3	2	12
Doubling	20	16	2	2	0	4
Throw on D	2	0	2	0	0	2
Grip	5	2	3	0	0	3
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	311	217	67	27	7	101

TBSR	0,6977
FBSR	0,2154
MBSR	0,0868
ABSR	0,0225
BSER	0,3248

Precision	0,7457
Recall	0,6977

Green Hills of Tyrol

Scan aus Scots Guards I

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	12	0	2	1	3
Abschnittsbeginn/-ende	4	1	2	1	0	3
Wiederholungszeichen	4	1	0	3	0	3
Notenhöhe	77	22	35	20	2	57
Notenlänge	77	37	20	20	2	42
Ausrichtung	58	43	2	13	2	17
Punktierung	10	5	0	5	0	5
Einzelne Grace Note	25	11	6	8	0	14
Doubling	20	17	2	1	2	5
Throw on D	3	0	2	1	0	3
Grip	5	0	3	2	0	5
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	303	153	74	76	9	159

TBSR	0,5050
FBSR	0,2442
MBSR	0,2508
ABSR	0,0297
BSER	0,5248

Precision	0,6483
Recall	0,5050

Scotland the Brave

direkte bww-grafik

150 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	16	0	0	0	0
Abschnittsbeginn/-ende	2	0	0	2	0	2
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	91	13	0	0	13
Notenlänge	104	96	8	0	0	8
Ausrichtung	77	77	0	0	0	0
Punktierung	17	17	0	0	0	0
Einzelne Grace Note	31	29	2	0	0	2
Doubling	29	26	0	3	0	3
Throw on D	4	4	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	3	3	0	0	0	0
Sonstige Verzierung	2	0	2	0	0	2
	407	376	26	5	0	31

TBSR	0,9238
FBSR	0,0639
MBSR	0,0123
ABSR	0
BSER	0,0762

Precision	0,9353
Recall	0,9238

Scotland the Brave

direkte bww-grafik

150 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	1	0	0	0	0
Taktstrich	16	16	0	0	0	0
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	82	22	0	0	22
Notenlänge	104	93	11	0	0	11
Ausrichtung	77	77	0	0	0	0
Punktierung	17	17	0	0	0	0
Einzelne Grace Note	31	29	2	0	0	2
Doubling	29	29	0	0	0	0
Throw on D	4	4	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	3	3	0	0	0	0
Sonstige Verzierung	2	0	2	0	0	0
	407	372	37	0	0	37

TBSR	0,9091
FBSR	0,0909
MBSR	0
ABSR	0
BSER	0,0909

Precision	0,9091
Recall	0,9091

Scotland the Brave

direkte bww-grafik

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	16	0	0	0	0
Abschnittsbeginn/-ende	2	0	0	2	0	2
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	99	5	0	0	5
Notenlänge	104	100	4	0	0	4
Ausrichtung	77	76	0	0	0	0
Punktierung	17	17	0	0	0	0
Einzelne Grace Note	31	29	2	0	0	2
Doubling	29	29	0	0	0	0
Throw on D	4	4	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	3	0	3	0	0	3
Sonstige Verzierung	2	0	2	0	0	2
	407	388	17	2	0	19

TBSR	0,9509
FBSR	0,0418
MBSR	0,0049
ABSR	0
BSER	0,0467

Precision	0,9579
Recall	0,9509

Scotland the Brave

direkte bww-grafik

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	16	0	0	0	0
Abschnittsbeginn/-ende	2	2	0	2	0	2
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	91	13	0	0	13
Notenlänge	104	100	4	0	0	4
Ausrichtung	77	76	0	0	0	0
Punktierung	17	17	0	0	0	0
Einzelne Grace Note	31	29	2	0	0	2
Doubling	29	29	0	0	0	0
Throw on D	4	4	0	0	0	0
Grip	5	5	0	0	0	0
Taorluath	3	3	0	0	0	0
Sonstige Verzierung	2	2	2	0	0	2
	407	387	22	2	0	24

TBSR	0,9484
FBSR	0,0541
MBSR	0,0049
ABSR	0
BSER	0,059

Precision	0,9461
Recall	0,9484

Scotland the Brave

Scan aus bww-Ausdruck

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	3	1	0	0	1
Kreuzchen	8	5	3	0	0	3
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	11	3	2	0	5
Abschnittsbeginn/-ende	2	1	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	80	20	4	0	24
Notenlänge	104	84	16	4	0	20
Ausrichtung	77	75	0	2	0	2
Punktierung	17	16	0	1	1	2
Einzelne Grace Note	31	27	3	1	1	5
Doubling	29	26	2	1	1	4
Throw on D	4	1	3	0	0	3
Grip	5	2	3	0	0	3
Taorluath	3	2	1	0	0	1
Sonstige Verzierung	2	0	2	0	0	2
	407	333	59	15	3	77

TBSR	0,8182
FBSR	0,1450
MBSR	0,0369
ABSR	0,0074
BSER	0,1892

Precision	0,8430
Recall	0,8182

Scotland the Brave

Scan aus bww-Ausdruck

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	3	1	0	0	1
Kreuzchen	8	3	4	1	0	5
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	11	3	2	0	5
Abschnittsbeginn/-ende	2	1	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	71	32	1	1	34
Notenlänge	104	83	20	1	1	22
Ausrichtung	77	73	0	4	0	4
Punktierung	17	16	0	1	1	2
Einzelne Grace Note	31	27	4	0	2	6
Doubling	29	26	3	0	1	
Throw on D	4	3	1	0	0	1
Grip	5	2	3	0	0	3
Taorluath	3	1	2	0	0	2
Sonstige Verzierung	2	0	2	0	0	2
	407	320	77	10	6	93

TBSR	0,7862
FBSR	0,1892
MBSR	0,0246
ABSR	0,0147
BSER	0,2285

Precision	0,7940
Recall	0,7862

Scotland the Brave

Scan aus Scots Guards I

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	1	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	15	13	0	2	0	2
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	59	34	8	3	45
Notenlänge	104	57	36	8	3	47
Ausrichtung	77	49	17	8	3	28
Punktierung	11	1	0	10	0	10
Einzelne Grace Note	27	15	10	1	1	12
Doubling	33	28	3	2	0	5
Throw on D	4	2	1	1	0	2
Grip	5	5	0	0	0	0
Taorluath	3	1	1	1	0	2
Sonstige Verzierung	1	0	1	0	0	1
	391	233	104	41	10	155

TBSR	0,5959
FBSR	0,2660
MBSR	0,1049
ABSR	0,0256
BSER	0,3964

Precision	0,6715
Recall	0,5959

Scotland the Brave

Scan aus Scots Guards I

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	15	12	0	3	1	4
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	42	50	12	2	64
Notenlänge	104	41	51	12	2	65
Ausrichtung	77	46	23	8	2	33
Punktierung	11	2	0	9	0	9
Einzelne Grace Note	27	13	14	0	1	15
Doubling	33	24	2	7	0	9
Throw on D	4	0	3	1	0	4
Grip	5	2	3	0	0	3
Taorluath	3	3	0	0	0	0
Sonstige Verzierung	1	0	1	0	0	1
	391	191	148	52	8	208

TBSR	0,4885
FBSR	0,3785
MBSR	0,1330
ABSR	0,0205
BSER	0,5320

Precision	0,5504
Recall	0,4885

Scotland the Brave

direkte bww-grafik

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	8	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	16	0	0	0	0
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	86	18	0	0	18
Notenlänge	104	86	18	0	0	18
Ausrichtung	77	77	0	0	0	0
Punktierung	17	17	0	0	0	0
Einzelne Grace Note	31	25	2	4	0	6
Doubling	29	29	0	0	0	0
Throw on D	4	0	4	0	0	4
Grip	5	5	0	0	0	0
Taorluath	3	3	0	0	0	
Sonstige Verzierung	2	0	2	0	0	2
	407	358	45	4	0	49

TBSR	0,8796
FBSR	0,1106
MBSR	0,0098
ABSR	0
BSER	0,1204

Precision	0,8883
Recall	0,8796

Scotland the Brave

Scan aus bww Ausdruck

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	8	6	2	0	0	2
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	12	1	3	1	5
Abschnittsbeginn/-ende	2	1	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	74	22	8	2	32
Notenlänge	104	82	14	8	2	24
Ausrichtung	77	71	1	5	1	7
Punktierung	17	13	1	3	0	4
Einzelne Grace Note	31	22	7	2	1	10
Doubling	29	23	3	3	0	6
Throw on D	4	2	2	0	0	2
Grip	5	1	4	0	0	4
Taorluath	3	1	2	0	0	2
Sonstige Verzierung	2	0	2	0	0	2
	407	312	63	32	7	102

TBSR	0,7666
FBSR	0,1548
MBSR	0,0786
ABSR	0,0172
BSER	0,2506

Precision	0,8168
Recall	0,7666

Scotland the Brave

Scan aus Scots Guards I

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	15	13	0	2	0	2
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	104	57	38	9	2	49
Notenlänge	104	51	44	9	2	55
Ausrichtung	77	54	20	3	2	25
Punktierung	11	5	0	6	1	7
Einzelne Grace Note	27	17	7	3	0	10
Doubling	33	31	2	0	0	2
Throw on D	4	1	2	1	0	3
Grip	5	2	3	0	0	3
Taorluath	3	1	2	0	0	2
Sonstige Verzierung	1	0	1	0	0	
	391	238	120	33	7	160

TBSR	0,6087
FBSR	0,3069
MBSR	0,0844
ABSR	0,0179
BSER	0,4092

Precision	0,6521
Recall	0,6087

Swollow's Tail

Scan aus The Moidart Collection 1

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	3	1	0	0	1
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	0	1	0	1
Taktstrich	14	14	0	0	1	1
Abschnittsbeginn/-ende	4	3	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	118	46	55	17	2	74
Notenlänge	118	82	20	16	2	38
Ausrichtung	108	87	7	12	2	21
Punktierung	0	0	0	0	1	1
Einzelne Grace Note	50	24	12	14	0	26
Doubling	8	4	2	2	0	4
Throw on D	0	0	0	0	0	0
Grip	0	0	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	4	0	3	1	0	4
	429	263	101	63	8	172

TBSR	0,6131
FBSR	0,2354
MBSR	0,1469
ABSR	0,0186
BSER	0,4009

Precision	0,7070
Recall	0,6131

Swollow's Tail

Scan aus The Moidart Collection 1

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	14	14	0	0	0	0
Abschnittsbeginn/-ende	4	3	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	118	42	53	23	3	79
Notenlänge	118	66	29	23	3	55
Ausrichtung	108	79	9	20	2	31
Punktierung	0	0	0	0	9	9
Einzelne Grace Note	50	24	11	15	0	26
Doubling	8	7	0	1	0	1
Throw on D	0	0	0	0	0	0
Grip	0	0	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	4	1	2	1	0	3
	429	240	106	83	17	206

TBSR	0,5594
FBSR	0,2471
MBSR	0,1935
ABSR	0,0396
BSER	0,4802

Precision	0,6612
Recall	0,5594

Swollow's Tail

Scan aus The Moidart Collection 1

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	4	0	0	0	0
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	0	0	0	0
Taktstrich	14	13	1	0	0	1
Abschnittsbeginn/-ende	4	3	1	0	0	1
Wiederholungszeichen	0	0	0	0	0	0
Notenhöhe	118	30	47	41	31	119
Notenlänge	118	65	12	41	31	84
Ausrichtung	108	65	2	41	31	74
Punktierung	0	0	0	0	0	0
Einzelne Grace Note	50	24	10	16	0	26
Doubling	8	6	1	1	0	2
Throw on D	0	0	0	0	0	0
Grip	0	0	0	0	0	0
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	4	0	4	0	0	4
	429	210	78	140	93	311

TBSR	0,4895
FBSR	0,1818
MBSR	0,3263
ABSR	0,2168
BSER	0,7249

Precision	0,5512
Recall	0,4895

Am Porst Cròm

Scan aus unbekannter Quelle

250x250 Neuronen mit Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	0	3	1	0	4
Kreuzchen	0	0	0	0	1	1
Taktbezeichnung	1	0	1	0	0	1
Taktstrich	16	15	0	1	2	3
Abschnittsbeginn/-ende	2	1	1	0	1	2
Wiederholungszeichen	2	1	0	1	1	2
Notenhöhe	138	68	53	17	13	83
Notenlänge	138	55	56	27	13	96
Ausrichtung	132	59	46	25	13	84
Punktierung	2	0	0	2	8	10
Einzelne Grace Note	45	9	9	0	0	9
Doubling	3	3	0	0	0	0
Throw on D	4	0	4	0	0	4
Grip	3	0	3	0	0	3
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	0	0	0	0	0	0
	490	211	176	74	52	302

TBSR	0,4306
FBSR	0,3592
MBSR	0,1510
ABSR	0,1061
BSER	0,6163

Precision	0,4806
Recall	0,4306

Am Porst Cròm

Scan aus unbekannter Quelle

250x250 Neuronen ohne Notenlinien

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	0	2	2	0	4
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	0	1	0	1
Taktstrich	16	15	0	1	4	5
Abschnittsbeginn/-ende	2	2	0	0	0	0
Wiederholungszeichen	2	2	0	0	0	0
Notenhöhe	138	54	66	18	12	96
Notenlänge	138	58	62	18	12	92
Ausrichtung	132	54	55	23	8	86
Punktierung	2	0	1	1	9	
Einzelne Grace Note	45	5	15	25	1	41
Doubling	3	1	2	0	1	3
Throw on D	4	0	4	0	0	
Grip	3	1	2	0	0	2
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	0	0	0	0	0	0
	490	192	209	89	47	345

TBSR	0,3918
FBSR	0,4265
MBSR	0,1816
ABSR	0,0959
BSER	0,7041

Precision	0,4286
Recall	0,3918

Am Porst Cròm

Scan aus unbekannter Quelle

250x250 Neuronen ohne künstliche Daten

	NEBS	NTBS	NFBS	NMBS	NABS	NBSE
Notenschlüssel	4	1	0	3	0	3
Kreuzchen	0	0	0	0	0	0
Taktbezeichnung	1	0	0	1	0	1
Taktstrich	16	15	1	0	3	4
Abschnittsbeginn/-ende	2	1	1	0	0	1
Wiederholungszeichen	2	0	1	1	0	2
Notenhöhe	138	50	56	32	11	99
Notenlänge	138	62	44	32	11	87
Ausrichtung	132	63	38	31	11	80
Punktierung	2	0	0	2	4	6
Einzelne Grace Note	45	12	6	28	0	34
Doubling	3	2	1	0	0	1
Throw on D	4	0	3	1	0	4
Grip	3	0	3	0	0	3
Taorluath	0	0	0	0	0	0
Sonstige Verzierung	0	0	0	0	0	0
	490	206	154	131	40	325

TBSR	0,4204
FBSR	0,3143
MBSR	0,2673
ABSR	0,0816
BSER	0,6633

Precision	0,515
Recall	0,4204

Abbildungsverzeichnis

Literaturverzeichnis

[1] *Scots Guards - Volume 1: Standard Settings of Pipe Music.* Patterson Publications, 1954.

[2] Bainbridge, David und Bell, Tim: *The Challenge of Optical Music Recognition.* Computers and the Humanities, 35(2):95–121, 2001.

[3] Bellini, Pierfrancesco, Ivan Bruno und Paolo Nesi: *Assessing Optical Music Recognition Tools.* Computer Music Journal, 31(1):68–93, März 2007.

[4] Brown, Ian: *From Tartan to Tartanry: Scottish Culture, History and Myth: Scottish Culture, History and Myth.* Edinburgh University Press, 2010.

[5] Claus Weihs, Dietmar Jannach, Igor Vatolkin Guenter Rudolph: *Music Data: Beyond the Signal Level.* In: *Music Data Analysis*, Chapman & Hall/CRC Computer Science & Data Analysis, Seiten 197–215. CRC Press, September 2016.

[6] Gunn, William: *A Bibliography for Bagpipe Music.* Gunn, William, 1848.

[7] Homenda, Władysław und Luckner, Marcin: *Automatic Recogniton of Music Notation Using Neural Networks.* In: *Procedings of the International Conference On AI and Systems IEEE AIS'04*, Seite 74–80, 2004.

[8] Kassler, Michael und Howard, Dennis Pruslin und David Stewart Prerau: *Optical Character-Recognition of Printed Music: A Review of Two Dissertations.* Perspectives of New Music, 11(1):250, 1972.

[9] MacDonald, Allan: *The Moidart Collection Volumn 1.* 1989.

[10] Music Show Scotland. `https://musicshowscotland.de/wp-content/uploads/2015/03/dudelsack_names-1.jpg`, aufgerufen am 24.Mai 2017.

[11] Pruslin, Dennis Howard: *Automatic recognition of sheet music.* phdthesis, Massachusetts Institute of Technology, Department of Mathematics, 1966.

[12] Rojas, Raul: *Neural Networks - A Systematic Introduction.* Springer, 1996.

[13] Rossant, Florence: *A global method for music symbol recognition in typeset music sheets.* Pattern Recognition Letters, 23(10):1129–1141, aug 2002.

[14] Wickstrom, Doug: *Bagpipe Player Software.* `http://www3.telus.net/public/dougwick/`, aufgerufen am 14.Mai 2017.

Eidesstattliche Versicherung

_____ _____

Name, Vorname Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/Masterarbeit* mit dem Titel

_____

_____

_____

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

_____ _____

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

_____ _____

Ort, Datum Unterschrift