Ansätze zur datengetriebenen Transkription einstimmiger Jazzsoli

Stefan Balke, Christian Dittmar, Meinard Müller

International Audio Laboratories Erlangen, Email: stefan.balke@audiolabs-erlangen.de

Die Transkription von Musiksignalen in symbolische Notendarstellungen stellt eine zentrale Fragestellung in der automatischen Musikverarbeitung dar. Die Herausforderung besteht dabei in der großen Variabilität der Tonerzeugung von Melodieinstrumenten, sowie der Mehrdeutigkeit im harmonischen Zusammenspiel mehrerer Instrumente. Im Zuge der zunehmenden Popularität von Deep Learning geht der Trend weg von modell- und regelbasierten Ansätzen hin zu datengetriebenen Transkriptionsverfahren. Exemplarisch zeigen wir anhand eines qualitativ hochwertigen Korpus manuell transkribierter, einstimmiger Jazzsoli, welche Deep Learning Architekturen für eine solche Aufgabenstellung als geeignet erscheinen. Dabei formulieren wir das Transkriptionsproblem als Klassifikationsaufgabe, bei welcher die Aktivierungen der verschiedenen Tonhöhen über die Zeit als Zielvorgabe für das Training verwendet werden. Ein besonderes Augenmerk richten wir in diesem Beitrag auf den Vergleich verschiedener Merkmalsdarstellungen und häufig in der Literatur verwendeter Metaparameter (z. B. die Netzwerktiefe oder der zeitliche Kontext).

Einleitung

Musiktranskription kann als der inverse Prozess zum Musizieren angesehen werden, bei welchem aus einer gegebenen Musikaufnahmen eine symbolische Notation ausgewählter Stimmen zurückgewonnen wird [3]. Besonders in der Jazzpädagogik spielt das manuelle Transkribieren von Soli durch den Musiker eine wichtige Rolle beim Studium von Improvisation. Allerdings erfordert dieser Prozess sehr viel Zeit und Konzentration und wird deshalb nur auf ausgewählten Beispielen durchgeführt. Für Musikwissenschaftler könnte eine zuverlässige, automatische Methode zur Musiktranskription den Weg für groß angelegte Korpusstudien in der Jazzforschung erleichtern. Dabei besteht allerdings die Problematik, dass automatische Musiktranskription als eine der schwierigsten Aufgabenstellungen im Forschungsfeld des Music Information Retrieval (MIR) gilt [9]. Obwohl es in den letzten Jahren im Bereich der automatischen Musiktranskription kontinuierliche Fortschritte gab [3, 7], kommen die algorithmischen Ergebnisse noch nicht an die Qualität von manuellen Transkriptionen heran.

In dieser Arbeit stellen wir, ähnlich wie in [2], ein datengetriebenes Verfahren vor, um den Melodieverlauf der Solostimme von bekannten Jazzaufnahmen zu extrahieren. Insbesondere die Überlagerung von Solist und den begleitenden Instrumenten (typischerweise eine Rhythmusgruppe bestehend aus Bass, Schlagzeug, Klavier) erschwert diese Aufgabe. Abbildung 1 zeigt schematisch zwei Hauptaspekte, die bei einer solchen automatischen Transkription von Bedeutung sind. Basierend auf einer Zeit-Frequenz Darstellung, die direkt aus dem Audiosignal extrahiert werden kann, gilt es sowohl die zeitliche Aktivität des zu analysierenden Instrumentes, als auch den Verlauf der Tonhöhe zu schätzen.

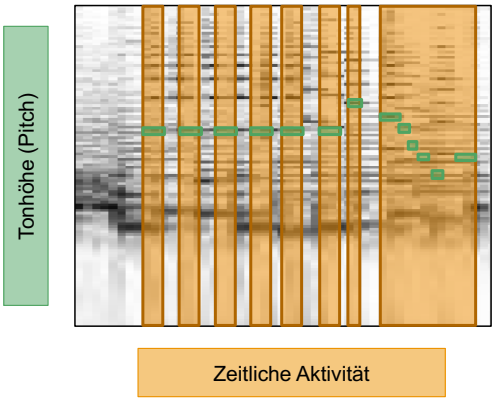

Abbildung 1: Illustration der zwei Hauptaspekte der automatischen Musiktranskription: Schätzung der zeitlichen Aktivität und Ermittlung der zugehörigen Tonhöhe.

Datengetriebene Transkription

In unserem Ansatz zur datengetriebenen Transkription benutzen wir künstliche neuronale Netzwerke (Deep Neural Networks, DNN), die trainiert werden, um die eben benannten Teilaufgaben simultan zu bewerkstelligen. Das Trainieren von DNNs erfolgt mittels überwachten Lernverfahren. Die dazu notwendigen Beispielpaare von Musikaufnahme und Zieltranskription entnehmen wir der *Weimar Jazz Database* (WJD) [10]. Die WJD enthält 456 Transkriptionen berühmter, improvisierter Jazzsoli, die von Musikwissenschaftlern annotiert und auf Richtigkeit geprüft wurden. Die betrachteten Soli wurden ausschließlich auf einstimmigen Instrumenten (z. B. Saxophon, Trompete oder Posaune) gespielt, werden aber in der Regel von einer Rhythmusgruppe begleitet. Zusammen mit den zugehörigen Audioaufnahmen ergibt sich ein annotierter Datenbestand mit einer Gesamtspielzeit von ca. 14 Stunden. Die durchschnittliche Dauer eines Solos beträgt ca. 2 Minuten, in dem der Solist im Schnitt in 60% der Gesamtlänge aktiv ist. Die Verteilung der Tonhöhen für alle Soli ist in Abbildung 3 aufgezeigt. Für das Training des DNN partitionieren wir den Gesamtdatensatz in einen Trainings- (ca. 8 Stunden), Validierungs- (ca. 3 Stunden) und Testdatensatz (ca. 3 Stunden).

Die Architektur unseres DNNs ist in Abbildung 2 dargestellt. Ähnliche Architekturen wurden bereits für die automatische Klaviertranskription verwendet, z. B. in [6].

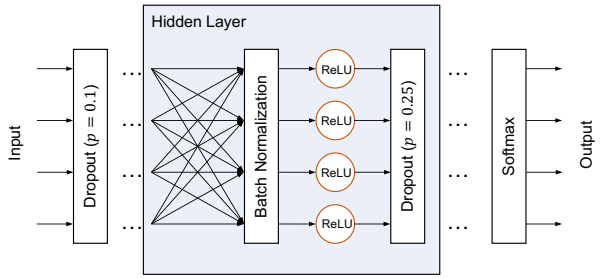

Abbildung 2: Darstellung der Netzwerkarchitektur. Der blau hinterlegte Block umrahmt einen Hidden Layer, der je nach Tiefe des Netzwerks wiederholt eingefügt wird.

Metaparameter	**Werte**
Frequenzaufteilung (#Dim.)	{lin. (2048), **log.** (88), mel (180)}
Amplitudenkompression	{keine Kompr., **log. Kompr**}
Zeitlicher Kontext (in Frames)	$\{0, \pm\mathbf{1}, \pm2, \pm4, \pm8\}$
Anzahl Hidden Layer	$\{1, 2, \mathbf{3}, 4, 5\}$

Tabelle 1: Übersicht über die verwendeten Metaparameter. Die Parameterkombination, die in unserem Evaluations-Szenario zu den besten Resultaten führte ist fett markiert.

Die Aufgabe des Netzwerkes besteht darin, die gegebene Zeit-Frequenz Darstellung der Musikaufnahmen auf 88 Tonhöhen im Bereich von A0 = 27,5 Hz bis C8 = 4186 Hz abzubilden. Zusätzlich benutzen wir noch eine 89. Klasse, um die Inaktivität des Solisten zu berücksichtigen. Da die Soli in der WJD mit einstimmigen Instrumenten gespielt wurden, nehmen wir in unserem Ansatz an, dass immer nur eine der 89 Klassen aktiv sein kann. Als Repräsentation der Musikaufnahmen am *Input Layer* des Netzwerks verwenden wir verschiedene Zeit-Frequenz Merkmale (z. B. extrahiert mittels *short-time Fourier transform*). Im einfachsten Fall verarbeitet das Netzwerk einzelne Merkmalsvektoren (Frames), allerdings sehen wir auch die Möglichkeit vor, diese mit weiteren Frames zur Erfassung des vergangenen und zukünftigen Zeitkontextes zu konkatenieren. Diese Merkmale werden im Anschluss in einer Kaskade von *Hidden Layers* weiterverarbeitet (blauer Teil in Abbildung 2). Durch die Aneinanderreihung mehrerer Hidden Layers erhält man tiefere Netzwerke mit mehr freien Parametern, die gelernt werden können. Nach dem letzten Hidden Layer verwenden wir eine *Softmax* Funktion, um am Ausgang des Netzwerkes eine Wahrscheinlichkeitsdichtefunktion zu approximieren. Die Klasse mit der höchsten Wahrscheinlichkeit entspricht dann unserer Prädiktion der Tonhöhe (oder Inaktivität) im momentan vom Netzwerk verarbeiteten Merkmalsvektor. Zusätzlich verwenden wir *Dropout* [13] und *Batch Normalization* [5] in unserem DNN, um den Trainingsprozess zu verbessern und eine Überanpassung an die Trainingsdaten zu vermeiden. Zur Realisierung des Trainings verwenden wir eine Kombination der Python Pakete `keras` [4] und `librosa` [8]. Für die anschließende Evaluation verwenden wir die *Pitch Accuracy*, *Voicing Recall*, *Voicing False Alarm* und *Overall Accuracy* für die Bewertung automatischer Melodieextraktionsverfahren. Hierzu greifen wir aus Implementationen aus `mir-eval` [11] zurück.

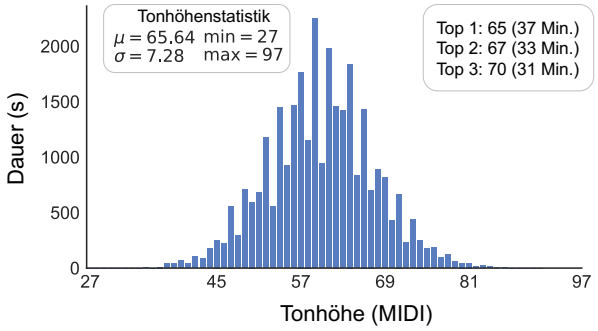

Abbildung 3: Verteilung der verschiedenen Tonhöhen in den Solotranskriptionen der WJD. (Die MIDI-Note 69 entspricht dem Ton A4 in MIDI Notation).

Neben den notwendigen annotierten Trainingspaaren müssen vor dem eigentlichen Training der DNNs eine Reihe von Metaparametern festgelegt werden (z. B. die Anzahl der Hidden Layer). Tabelle 1 listet die Metaparameter auf, deren Einfluss wir in den nun folgenden Experimenten weiter untersuchen.

Experimente und Diskussion

Für jede der 150 Kombinationen von Metaparametern wird jeweils ein Modell trainiert und mittels fünffacher Kreuzvalidierung auf den Validierungsdaten ausgewertet. In den Experimenten hat sich gezeigt, dass die Eingangsmerkmale mit einer linearen Frequenzachse die besten Ergebnisse liefern. Sobald allerdings die Amplituden logarithmisch komprimiert werden, sind lineare und logarithmische Frequenzaufteilung gleichauf. Da letztere allerdings eine wesentlich kleinere Dimensionalität aufweist (2048 vs. 88), ziehen wir die logarithmische Aufteilung der linearen vor. Durch Zugabe von zeitlichem Kontext aus den jeweils benachbarten Frames (ca. 300 ms) konnten wir eine Verbesserung um ca. 10 % in der Overall Accuracy feststellen. Hinzunahme von größerem zeitlichen Kontext hatte in unserem Szenario keinen Effekt auf die Ergebnisse. Als Netzwerktiefe hat sich eine Anzahl von 3 Hidden Layers als ausreichend herausgestellt.

Zusammenfassend lässt sich feststellen, das eine Kombination aus logarithmischer Frequenzachse und logarithmischer Kompression der Eingangsmerkmale, ein zeitlicher Kontext von ±1 Frames (entspricht ca. 300 ms), sowie einer Netzwerktiefe von 3 Hidden Layers die besten Ergebnisse nach fünffacher Kreuzvalidierung liefert (*Overall Accuracy* = 0.7). Als Vergleichsmethode zu unserem DNN verwenden wir den bekannten, regelbasierten Ansatz *Melodia* [12]. Dieser Algorithmus beruht auf einer speziellen Salienzfunktion, in der die Grundfrequenz des dominanten Instrumentes (in unserem Fall die des Solisten), mit Hilfe von subharmonischer Summation verstärkt wird. Mittels einer Reihe von Regeln wird im Anschluss aus dieser Darstellung der Melodieverlauf des Solisten geschätzt. *Melodia* erreicht auf diesem Datensatz eine *Overall Accuracy* von lediglich 0.31. Insbesondere die Aktivitätsschätzung ist mit einem *Voicing False Alarm* von 0.98 sehr schlecht—im Grunde wurde fast jeder Frame als aktiv angenommen. Weitere Details zu den

Ergebnissen sind in [1, Appendix C] zu finden.

Im Hinblick auf die große Diskrepanz zwischen den Ergebnissen des regelbasierten *Melodia* Algorithmus und unseres datengetriebenen Ansatzes wollen wir betonen, dass der direkte Vergleich der beiden Ansätze nicht ganz fair ist. Im Gegensatz zum DNN besitzt *Melodia* keinerlei Kenntnisse über die Verteilung der Daten. Zudem wurde in mehreren Studien gezeigt, dass die in *Melodia* verwendeten Konzepte auf einer Vielzahl von Musikstilen anwendbar und erweiterbar sind. Hingegen funktioniert unser Ansatz zur datengetriebenen Solotranskription für Jazzmusik, allerdings wird er nicht ohne weitere annotierte Trainingsdaten auf andere Musikstile anwendbar sein. In Zukunft wollen wir in diesem Zusammenhang untersuchen, wie traditionelle Methoden mittels Zugabe von Informationen über die Verteilung der Trainingsdaten verbessert werden können. Weiterhin wollen wir untersuchen, wie traditionelle Transkriptionsmethoden mit den datengetriebenen Methoden verbunden werden könnten, um damit die Generalisierbarkeit der DNNs zu erhöhen.

Danksagung
Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft unterstützt (DFG MU 2686/11-1). Die International Audio Laboratories Erlangen sind ein Zusammenschluss der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) und dem Fraunhofer-Institut für Integrierte Schaltungen IIS. Die Autoren bedanken sich beim Regionalen Rechen Zentrum Erlangen (RRZE) für die Bereitstellung und Unterstützung bei der Nutzung der Computercluster.

Literatur

[1] Balke, S. (2018): Multimedia Processing Techniques for Retrieving, Extracting, and Accessing Musical Content. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU).

[2] Balke, S., Dittmar, C., Abeßer, J. & Müller, M.: Data-Driven Solo Voice Enhancement for Jazz Music Retrieval. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 196–200 (2017).

[3] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H. & Klapuri, A.: Automatic music transcription: challenges and future directions. In: Journal of Intelligent Information Systems (2013), **41**, 3: 407–434. URL `http://dx.doi.org/10.1007/s10844-013-0258-3`.

[4] Chollet, F. et al. (2015): Keras. `https://keras.io`.

[5] Ioffe, S. & Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the International Conference on International Conference on Machine Learning (ICML), 448–456 (2015).

[6] Kelz, R., Dorfer, M., Korzeniowski, F., Böck, S., Arzt, A. & Widmer, G.: On the Potential of Simple Framewise Approaches to Piano Transcription. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 475–481. New York City, USA (2016).

[7] Klapuri, A. P. & Davy, M. [Hrsg.] : Signal Processing Methods for Music Transcription. Springer, New York (2006).

[8] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E. & Nieto, O.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th Python in Science Conference, 18–25 (2015).

[9] Müller, M.: Fundamentals of Music Processing. Springer Verlag (2015).

[10] Pfleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.-G. & Burkhart, B. [Hrsg.] : Inside the Jazzomat. New perspectives for jazz research. Schott Campus, Mainz, Germany (2017).

[11] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D. & Ellis, D. P. W.: MIR_EVAL: A Transparent Implementation of Common MIR Metrics. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR), 367–372. Taipei, Taiwan (2014).

[12] Salamon, J. & Gómez, E.: Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics. In: IEEE Transactions on Audio, Speech, and Language Processing (2012), **20**, 6: 1759–1770.

[13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: Journal of Machine Learning Research (2014), **15**: 1929–1958.