JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Matthias Dorfer

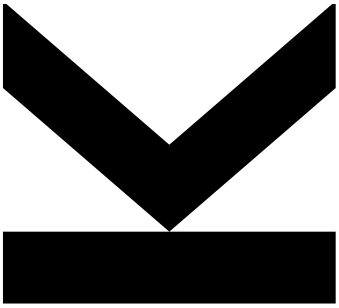
Submitted at
Department of Computational Perception

Supervisor and
First Examiner
Gerhard Widmer

Second Examiner
Douglas Eck

October 2018

Multimodal Deep Representation Learning and its Application to Audio and Sheet Music



Doctoral Thesis
to obtain the academic degree of
Doktor der technischen Wissenschaften
in the Doctoral Program
Technische Wissenschaften

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696

Statutory Declaration

I hereby declare that the thesis submitted is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references. This printed thesis is identical with the electronic version submitted.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe. Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.

Abstract

This thesis is about multimodal deep representation learning and its application to audio and sheet music. Multimodal deep learning in general could be described as learning task-specific representations from two or potentially more input modalities at the same time. What kind of representations a model learns mainly depends on the given training data and the task that is addressed, including its respective optimization target.

In the first part of my thesis, the data at hand are images of sheet music and their corresponding music audio. Three different machine learning paradigms are employed to address Music Information Retrieval (MIR) problems involving audio and sheet music, with multimodal convolutional neural networks. In particular, the thesis presents (1) supervised function approximation for score following directly in sheet music images, (2) multimodal joint embedding space learning for piece identification and offline audio – score alignment, and (3) deep reinforcement learning again addressing the task of score following in sheet music. All three approaches have in common that they are built on top of multimodal neural networks that learn their behavior purely from observations presented during training.

To train such networks a suitable and large enough dataset is required. As such data was not available when I started working on the thesis, I have collected a free, large-scale, multimodal audio–sheet music dataset, with complete and detailed alignment ground-truth at the level of individual notes. In total the dataset covers 1,129 pages of music, which is exactly the kind of data required to explore the potential of powerful machine learning models. The dataset, including my experimental code, is made freely available to foster further research in this area. With this new dataset I show that with the right combination of appropriate data and methods it is feasible to learn solutions for complex MIR-related problems entirely from scratch without the need for musically-informed hand-designed features.

In the second part of my thesis I take a step back from this concrete application and propose methodological extensions to neural networks in general, which are more broadly applicable beyond the domain of audio and sheet music. We revisit Canonical Correlation Analysis (CCA) and Linear Discriminant Analysis (LDA) – two methods from multivariate statistics – to extend their core ideas to allow for combination with deep neural networks. In the case of CCA, I show how to improve cross-modality retrieval via multimodal embedding space learning by back-propagating a ranking loss directly through the analytical projections of CCA. For LDA, I reformulate its central idea as an optimization target to train neural networks that produce discriminative, linearly separable latent representations useful for classification tasks such as object recognition.

To summarize, this thesis extends the application domain of multimodal deep learning to audio and sheet music-related MIR problems, proposes a novel audio – sheet music dataset, and adds two general methodological contributions to the field of deep learning.

Zusammenfassung

Diese Dissertation beschäftigt sich mit dem Erlernen abstrakter Repräsentationen mit Hilfe von *tiefen, multimodalen neuronalen Netzen (Multimodal Deep Learning)* und deren Anwendung auf Audioaufnahmen von Musikstücken und den korrespondierenden Notentexten. Multimodal bedeutet hierbei, dass während des Lernprozesses mehrere, verschiedene Eingangsmodalitäten simultan verwendet werden. Die Eigenschaften der erlernten Repräsentationen hängen dabei hauptsächlich von den zugrundeliegenden Trainingsdaten, sowie der zu lösenden Aufgabe und der Zielfunktion des Optimierungsprozesses ab.

Im ersten Teil meiner Dissertation bestehen diese multimodalen Daten aus Bildern von Notentexten und korrespondierenden Audioaufnahmen der abgebildeten Musikstücke. Diese Daten werden in drei unterschiedlichen Lernparadigmen verwendet, um beispielhafte Problemstellungen aus dem Bereich des *Music Information Retrieval* zu behandeln: (1) *Supervised Learning* wird verwendet, um die aktuelle Position einer abgespielten Musikaufnahme direkt im Notententext zu verfolgen. (2) spezielle multimodale Abbildungen (*Embeddings*) werden gelernt, um die effiziente Suche in Notentextdatenbanken, sowie die automatische Synchronisation von Musikaufnahmen mit Notentexten zu ermöglichen. (3) *Deep Reinforcement Learning* wird als alternatives Lernparadigma für die automatische Verfolgung der Notentextposition vorgestellt. Alle drei Ansätze haben gemein, dass die Funktionsweisen der multimodalen neuronalen Netze einzig aus den zugrundeliegenden Trainingsbeispielen erlernt werden.

Für das Trainieren dieser Netze ist ein ausreichend großer Trainingsdatensatz erforderlich. Zu Beginn dieser Arbeit war kein geeigneter Datensatz verfügbar, weshalb ich im Verlauf meiner Forschung eine frei verfügbare Datenbank mit Notentexten und korrespondierenden Musikaufnahmen zusammengestellt habe. Dieser Datensatz umfasst 1129 Seiten Notentext, die exakt auf Notenkopfebene mit der entsprechenden Audioaufnahme synchronisiert sind. Systematische Experimente auf diesem Datensatz zeigen, dass es möglich ist, Lösungen für komplexe musikbezogene Problemstellungen vollständig datengetrieben und daher mit sehr geringem Einsatz von domänenspezifischem Zusatzwissen zu erlernen.

Der zweite Teil dieser Arbeit verlässt diesen konkreten musikalischen Anwendungsbereich und stellt zwei methodische Erweiterungen für neuronale Netze vor. Diese Erweiterungen haben ihren Ursprung in der klassischen multivariaten Statistik: Der kanonischen Korrelationsanalyse und der linearen Diskriminanzanalyse. Ich greife in beiden Fällen die Kernkonzepte der jeweiligen Methode auf und adaptiere diese für die Kombination mit tiefen neuronalen Netzen. Im ersten Fall entwerfe ich einen differenzierbaren, auf der kanonischen Korrelationsanalyse basierenden *Layer* für neuronale Netze, der ein effizientes Erlernen von *Embeddings* über Modalitätsgrenzen hinweg ermöglicht (z.B. für die Suche von Bildern auf Basis einer textuellen Suchanfrage). Im zweiten Fall formuliere ich die lineare Diskriminanzanalyse als Zielfunktion für das Trainieren neuronaler Netze. Die daraus resultierenden Repräs-

sentationen sind linear separierbar und können daher für Klassifikationsprobleme (z.B. Objekterkennung) eingesetzt werden.

Zusammengefasst erweitert die vorliegende Dissertation das Anwendungsfeld von multimodalen neuronalen Netzen auf musikalische Problemstellungen im Kontext von Notentexten und korrespondierenden Audioaufnahmen, veröffentlicht einen neuen für diese Forschung notwendigen Datensatz und erweitert zwei Verfahren der klassischen multivariaten Statistik für deren Kombination mit tiefen neuronalen Netzen.

Acknowledgements

First of all, I would like to thank my supervisor Gerhard Widmer for giving me the opportunity to do my PhD in such a welcoming and nice environment, for all his trust to follow my research interests, for supporting me with visiting all the awesome conferences, and for being a great and inspiring teacher. Secondly, I warmly thank Doug Eck, my second examiner, for taking the time and effort to review my thesis and research. My special thanks go to Thomas Hoch, Bernhard Moser and Theodorich Kopetzky for their support with arranging the collaboration with the Institute of Computational Perception and of course also to the Software Competence Center Hagenberg for funding a substantial part of my PhD studies.

Doing a PhD on your own can be a frustrating and tiring task. Fortunately, I was very lucky and had a lot of great colleagues with who I ended up in plenty of fruitful and exciting collaborations. Thanks to Bernhard Lehner, Hamid Eghbal-Zadeh, Andreu Vall, Florian Henkel, Marko Tkalcic, Khaled Koutini, Filip Korzeniowski, Shreyan Chowdhury, Reinhard Sonnleitner, Christine Bauer, Markus Schedl, Bruce Ferwerda, Thassilo Gadermaier, Harald Frostel, Andreas Arzt, Florian Krebs, Rainer Kelz, Richard Vogl, Stefan Lattner, Peter Knees, Sebastian Böck, Stefan Balke, Jan Schlüter, and Carlos Eduardo Cancino-Chacon. These people and collaborations are one of the aspects I really enjoyed the most during my PhD studies.

I also want to thank Andreas Arzt, Harald Frostel, Florian Krebs, and Martin Preinfalk for accepting me as the fifth wheel in their four-person office starting from my first day in the lab; Jan Schlüter for sharing his great thesis template, for developing Lasagne (the basis of my research), and for sharing all his Theano skills with me; Rainer Kelz and Harald Frostel for building our GPU servers; Florian Henkel for being the best possible first master student to advise; Jan Hajič for initiating our collaboration leading to all the cool projects we did together; the three DCASE teams for their infinite stress resistance pushing our audio classification systems to belong to the best ones in the world; Claudia Kindermann for her support with all the organizational stuff; Harald Frostel for organizing his professional team-building events in Offensee; and finally Andreas Arzt for being the social heart (and event manager) of the institute and for taking all the effort to make the lab a wonderful place to work at.

I also want to take this opportunity to thank my wonderful parents and family for supporting me as long as I am able to remember. And last but definitely not least I want to thank Conny with all my heart for her endless support and patience.

The research reported in this thesis has been supported in part by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH. Part of the work was also supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme. (ERC Grant Agreement 670035, project CON ESPRESSIONE). The Tesla K40 used for some of the experiments was donated by the NVIDIA Corporation.

Contents

List of Figures

List of Tables

1 Introduction and Outline

This thesis is situated in the intersection of the fields machine learning and Music Information Retrieval (MIR) and is concerned with multimodal deep representation learning in the context of music audio and corresponding scores given as images of sheet music. Besides working on this concrete application scenario, I also propose methodological extensions to deep learning and neural networks in general, which are more broadly applicable beyond the domain of audio and sheet music. I will start by motivating this thesis from two different perspectives. The first is centered around the performer or the consumer of music and addresses the practical relevance and potential applications of my research. The second perspective looks at the tasks from a machine learner's point of view and is inspired solely by academic interest.

From an application point of view the research presented in this thesis originates in the rapidly growing, huge amount of digitally available music data. Such digital music collections comprise, for example, audio recordings, digitized images of sheet music, album covers and liner notes, and an increasing number of video clips. Facing these vast volumes of multimodal data makes it difficult and time consuming to manually manage such collections, and raises the need for support by intelligent systems (Balke, 2018). One important property of this kind of data is its multimodality, meaning that entities are represented by two or more of the mentioned modalities at the same time. The modality pair especially relevant for this thesis is audio recordings of musical performances and the corresponding scores of the respective pieces. One of the application in this domain is score following, the process of following a musical performance in a corresponding representation of the score. Assuming that we have such an automatic score following system, we can address tasks such as automatic page turning or – in the case of a very accurate score follower – even design agents that automatically accompany human musicians in a joint performance. We will take a close look at score following in Chapters 3 and 5 of this thesis. A second use cases, is cross-modality retrieval: Given a search query in one modality (e.g. an audio recording) the goal is to retrieve relevant items within a collection of documents given in a different modality (e.g. the corresponding score of the piece, or a video of a live performance). We will turn to cross-modality retrieval in Chapters 4 and 6.

These are appealing applications, and for music audio and computer-readable representations of the musical score (e.g. MIDI) there are already solutions that work robustly and flexibly enough to be useful in real world scenarios (Liem et al.,

2015; Arzt, 2016). However, the main drawback of these approaches is that they rely on symbolic, machine-readable music representations, which are rarely available and extremely time-consuming to prepare (Arzt, 2016). On the other hand, when working directly with images of printed sheet music, the tasks become considerably harder, and there are many open problems that need to be addressed to reach the performance of systems relying on symbolic scores.

That is the starting point of my research and this thesis where I will try to lift the limitations introduced when working with printed sheet music. I propose three different machine learning approaches that learn their behavior purely from observations presented during training time. The overall vision is to build machine learning systems that are capable of dealing with any kind of classical sheet music, out of the box, with no need for time consuming data preparation.

Before I continue with motivating my thesis from a machine learning point of view, I first want to note that this work is of course not the first attempt to address tasks such as score following or score–performance retrieval directly in images of sheet music (Dannenberg, 1984). However, what existing approaches have in common is the fact that they are based on hand-curated features and procedures (see e.g. (Müller, 2015) for an overview). When I started working on this thesis the seminal ImageNet paper by Krizhevsky et al. (2012) was long published and the neural network revival was already in full swing. Artificial neural networks, now also termed deep learning, had proven to be very powerful in other domains, and it became feasible to address tasks such as object recognition or image captioning directly from images (without any feature engineering) simply by optimizing a network with training examples. Having this success in mind and looking at the intersection of both fields, machine learning and MIR, there was one question most conspicuous to me: Is it possible to design machine learning systems that are capable of learning to solve complex, multimodal music-related tasks such as score following in an end-to-end fashion directly from the raw inputs (images and audio).

This is the essence of my research and this thesis. We will see in the remainder of this work that it comprises many difficult machine learning problems making it, besides being relevant to the MIR community, a problem of more general academic relevance.

1.1 Main Contributions

I briefly summarize the main contributions of my thesis below. A more detailed overview of my contributions is provided in the outline in Section 1.2.

1. As the central contribution of my thesis I propose to address audio – sheet music related MIR problems with multimodal convolutional networks. This has not been done before, and I successfully show, using three different machine learning paradigms, the potential of such models for the data and problems at hand. In particular, I cover (1) supervised function approximation for score following directly in sheet music images, (2) multimodal joint embedding space learning for piece identification and offline audio – score alignment, and (3) deep reinforcement learning again addressing the task of score following in sheet music. All three scenarios have in common that they are situated around multimodal representation learning on audio and sheet music images.

2. I propose and release a free, large-scale, multimodal audio–sheet music dataset, with complete and detailed alignment ground-truth at the level of individual notes. The dataset and the corresponding code is distributed to the research community in the hope of opening many sheet music related MIR tasks (in addition to the ones already outlined in this thesis) to state-of-the-art machine learning methods. As all methods presented in this thesis learn their behavior solely from examples presented for training this data set could be seen as the basis of my research.

3. Besides my research in the context of audio and sheet music I propose to reformulate classical Canonical Correlation Analysis (CCA) as a special purpose layer in multimodal neural networks for efficient cross-modal retrieval embedding space learning. This CCA-Layer is then already reused and successfully applied in subsequent research also presented in this thesis to learn retrieval embeddings spaces for audio excerpts and sheet music snippets.

4. I propose Deep Linear Discriminant Analysis (DeepLDA), a nonlinear, deep neural network extension to classic LDA as an alternative optimization target to categorical-cross-entropy for training classification networks. DeepLDA produces linearly separable latent representations which: (a) have low variance within the same class and (b) high variance between different classes. The learned representation are then useful in subsequent classification tasks such as object recognition in images.

5. I release my experimental code to foster further research and, especially for the sheet music case, to reduce initial hurdles for starting to work with this kind of data:

- A mutlimodal audio sheet music dataset (Chapter 2)
  https://github.com/CPJKU/msmd
- Audio – sheet music embedding space learning (Chapter 4)
  https://github.com/CPJKU/audio_sheet_retrieval
- The score following game (Chapter 5)
  https://github.com/CPJKU/score_following_game
- Canonical correlation analysis layer (Chapter 6)
  https://github.com/CPJKU/cca_layer
- Deep linear discriminant layer (Chapter 7)
  https://github.com/CPJKU/deep_lda

1.2 Thesis Outline

The remainder of this thesis is structured into three main parts. Part I focuses on the application of multimodal deep learning models to audio and sheet music images. Part II introduces two methodological extensions to neural networks which are both inspired by classical statistical models (Canonical Correlation Analysis and Linear Discriminant Analysis). The link between the two parts is the CCA layer proposed in Chapter 6 which I already applied successfully to audio - sheet music embedding space learning described in Chapter 4 of Part I. Finally, Part III concludes the thesis and provides an outline for future work.

1.2.1 Part I: Multimodal Deep Learning for Linking Audio and Sheet Music

Starting with the audio – sheet music applications, I will introduce three different machine learning paradigms all trained simultaneously on the two modalities at hand. All three approaches are data driven so before describing the methods in detail, I will first introduce a novel large scale dataset required to train such models.

Chapter 2, is based on Dorfer et al. (2018a). The three approaches introduced in Part I build on deep artificial neural networks and learn their behavior entirely from training observations. This implies that in order to arrive at models generalizing to unseen observations we first require a large and comprehensive training data set. At the time I started working on this thesis there was no such dataset available. Therefore, in this chapter I introduce MSMD, a multimodal audio sheet music dataset comprising 479 precisely annotated solo piano pieces by 53 composers, for a total of 1,129 pages of music and about 15 hours of aligned audio, which was synthesized from these scores. The dataset is delivered with fine grained annotations linking each musical

note onset event in the audio to its counterpart, the pixel coordinate of the corresponding note head, in the sheet music images. Given this large scale dataset, everything is ready to introduce and train the models.

Chapter 3, based on Dorfer et al. (2016b), is situated around supervised learning, striving to learn a function $f(x)$ which maps a given input observation x to its desired output y. In the supervised setting this mapping is learned from training examples given as a set of tuples (x, y). Taking this principle as a basis, I will show in this chapter how to utilize it to train a neural network that predicts, given a short excerpt of audio and an image of sheet music as an input, the most likely position of the audio in the score. Once such a multimodal localization network is trained it can be used for score following in sheet music images.

Chapter 4 is based on Dorfer et al. (2018a), Dorfer et al. (2017a), Dorfer et al. (2017b), and Dorfer and Widmer (2016b). In this chapter we are still in the regime of supervised function approximation, but follow a different learning paradigm, namely cross-modal embedding space learning. Given an excerpt of audio (for example one bar of music) and its corresponding snippet of sheet music we aim at learning two functions f and g, one for embedding the audios and one for embedding the sheet images. The overall goal is to arrive at a joint embedding space where semantically similar items of both modalities live close together and dissimilar items are placed far apart. Given such a space, I show how to utilize it for piece identification via cross-modality retrieval, and for offline audio – sheet music alignment via dynamic time warping using cross-modal distances in the learned embedding space.

Chapter 5 is based on Dorfer et al. (2018c) and proposes an alternative approach to score following directly in sheet music images, the same task as already addressed in Chapter 3. However, this chapter follows a completely different machine learning paradigm. I start this chapter by formulating score following as a multimodal Markov decision process, the mathematical foundation for sequential decision making. Given this formal definition, we address the score following task with state-of-the-art deep reinforcement learning algorithms. In contrast to the two previous approaches reinforcement learning does not rely on a supervised learning signal but instead provides us with the means to train agents which learn their behavior via interaction with their environment. For the present case, I term this environment the *Score Following Game*.

1.2.2 Part II: Deep Learning on Top of Classical Multivariate Statistics

In the second part of my thesis I present two methodological extensions to deep learning in general, where some of my proposals have already been successfully applied in Chapter 4 of Part I.

Chapter 6 is based on Dorfer et al. (2018d), and addresses the problem of cross-modality retrieval. Cross-modality retrieval encompasses retrieval tasks where the fetched items are of a different type than the search query, e.g., retrieving pictures relevant to a given text query. The state-of-the-art approach to cross-modality retrieval relies on learning a joint embedding space of the two modalities, where items from either modality are retrieved using nearest-neighbor search. In this chapter, I introduce a neural network layer based on Canonical Correlation Analysis that learns better embedding spaces by analytically computing projections that maximize correlation. I show the effectiveness of this layer in extensive retrieval experiments on different modality pairs such as image and text or audio and sheet music.

Chapter 7 is based on Dorfer et al. (2016d) and introduces Deep Linear Discriminant Analysis (*DeepLDA*) which learns linearly separable latent representations in an end-to-end fashion. The method is related to the one proposed in Chapter 6, as both build on well known methods from classical multivariate statistics and are trained on statistics such as covariance estimates of the given training observations. Classic LDA extracts features which preserve class separability and is used for dimensionality reduction for many classification problems. The central idea of this chapter is to put LDA on top of a deep neural network. This can be seen as a non-linear extension of classic LDA. I evaluate the approach on three different benchmark datasets (MNIST, CIFAR-10 and STL-10). DeepLDA produces competitive results on MNIST and CIFAR-10 and outperforms a network trained with categorical cross entropy (having the same architecture) on a supervised setting of STL-10.

1.2.3 Part III: Conclusions and Future Work

In the final part of my thesis I will draw a complete picture by summarizing the main findings and conclusions across all parts and chapters. I also try to provide a pathway through open research problems and directions which are in my opinion the most prominent and promising ones to tackle as next steps to advance the field.

1.3 Publications

The main chapters of this thesis build on the following publications:

- M. Dorfer, A. Arzt, and G. Widmer. Towards score following in sheet music images. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 789–795, New York City, USA, 2016b.
- M. Dorfer and G. Widmer. Towards end-to-end audio-sheet-music retrieval. In *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, Barcelona, Spain, 2016b.
- M. Dorfer, A. Arzt, and G. Widmer. Learning audio-sheet music correspondences for score identification and offline alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 115–122, Suzhou, China, 2017a.
- M. Dorfer, J. j. Hajič, A. Arzt, H. Frostel, and G. Widmer. Learning audio—sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval*, 1 (1):22–33, 2018a. doi:http://doi.org/10.5334/tismir.12.
- M. Dorfer, J. j. Hajič, and G. Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, Kyoto, Japan, 2017b.
- M. Dorfer, F. Henkel, and G. Widmer. Learning to listen, read, and follow: Score following as a reinforcement learning game (**Best Paper and Best Poster Award**). In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018c.
- M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, USA, 2016d.
- M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer. End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval (IJMIR)*, 7(2):117–128, 2018d.

For a complete list of publications published during (and before) my thesis work, please refer to the curriculum vitae at the very end of this document.

Part I

Multimodal Deep Learning for Linking Audio and Sheet Music

The first part of this thesis is concerned with applied multimodal deep learning in the context of audio and sheet music images. Multimodal deep learning in general could be described as learning task specific representations from two (or potentially more) input modalities at the same time. Which kind of representations a model learns mainly depends on two components, the given training data as well as the task that is addressed, including its respective optimization target. One representative application in this domain is cross-modality retrieval, where we aim at retrieving items of a different type than the search query, e.g., retrieving pictures relevant to a given text query. What makes multimodal problems difficult is the fact that we need to bridge the semantic gap between the two modalities at hand in example to suggest relevant items as a search result. Although I chose cross-modality retrieval as an illustrative use case for this introduction, we will see that there are various different forms of this general multimodal learning concept. In particular, I will show how to utilize three different machine learning paradigms, all building on top of multimodal deep learning, to successfully address audio – sheet music related MIR tasks. Figure 1.1 gives an overview of the three approaches introduced in the remainder of this part.

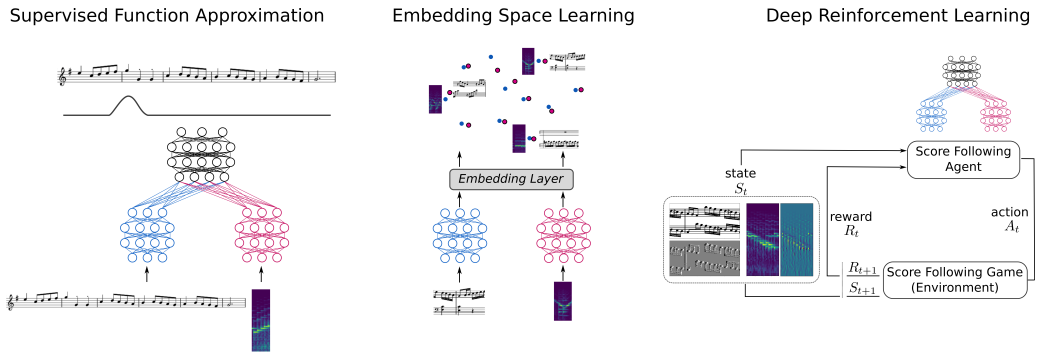


Figure 1.1: Illustration of the three machine learning paradigms covered in this thesis for addressing audio–sheet music related MIR tasks.

Supervised Function Approximation (Chapter 3). The first approach utilizes supervised function approximation to address score following in sheet music images. I will show how to apply concepts common for tasks such as object recognition to train neural networks that learn to predict, given a short excerpt of music audio, its most likely position it the respective image of sheet music. This can be also seen as a multimodal localization problem. From a machine learning point of view, we aim at learning a function $f(x)$ which maps a given input observation x to its desired output y. In the supervised case this mapping is learned from training examples given as a set of tuples (x, y).

Embedding Space Learning (Chapter 4). The second paradigm covered in this thesis is multimodal joint embedding space learning for cross-modality retrieval. When given an excerpt of audio (for example one bar of music audio) and its corresponding snippet of sheet music, we aim at learning two embedding projections f and g, one for embedding the audios and one for embedding the sheet images. This is a different learning paradigm but still a supervised problem as we need corresponding observations of both modalities for training the networks. The overall goal is to arrive at a joint embedding space which allows for a semantic comparison between entities across the two modalities. Once we have trained a network providing us with such a space, I will show how to utilize it for piece identification via cross-modality retrieval, and for offline audio – sheet music alignment via dynamic time warping on cross-modal distances in the learned embedding space.

Deep Reinforcement Learning (Chapter 5). Finally, we revisit the problem of score following in sheet music images, and I will show how to formulate it as a mutlimodal reinforcement learning task. The overall goal is to design reinforcement learning agents which learn to read scores, listen to the currently playing music, and then follow the performance along in the score. We will start by formulating score following as a multimodal Markov Decision Process, the mathematical foundation for sequential decision making. Given this formal definition, we have the basis to address the score following task with state-of-the-art deep reinforcement learning algorithms. From a machine learning perspective this is different compared to the previous approaches. Reinforcement learning does not rely on a supervised learning signal teaching the agent which action to take in a given state. Instead it enables us to train agents that learn their behavior solely from interaction with their environment[1].

The common ground of all three approaches is twofold: (1) The data they operate on are images of sheet music and spectrograms computed from music audio. (2) Their core component is a multimodal convolutional network learning a task-dependent behavior entirely from the observations presented during training. The latter implies that the driving force (besides the gradients for back-propagation of course (Rumelhart et al., 1986)) is, in all three cases, the data at hand. So before taking a closer look at the methodological contributions, I will first introduce and describe the kind of data used for training and evaluating the models.

[1] For the particular case of score following we will of course still need ground truth annotations linking scores and performances to compute the reward signal during training, i.e. the environment is aware of the actual ground truth and can distribute reward accordingly. I will elaborate on this in Chapter 5 in detail.

2 MSMD: A Multimodal Audio - Sheet Music Dataset

In this chapter, I introduce a Multimodal Sheet Music Dataset (MSMD) which is used to train and evaluate the methods described in Part I of this thesis. MSMD is a free, large-scale, multimodal audio–sheet music dataset, with complete and detailed alignment ground-truth at the level of individual notes. The dataset is built on top of the Mutopia Project[1], a collection of more than 2000 scores, collected under the Creative Commons licenses. This allows us to share and distribute the whole dataset to the research community. As the dataset was collected and prepared towards the end of my thesis, there are contributions contained in this thesis which were published and evaluated on different datasets. This applies especially to the method described in Chapter 3 on score following in sheet music images. In such cases, I will present the results of the original papers, but also reevaluate the methods when it is relevant, to provide a complete picture of the respective performances.

The MSMD dataset was described in the following publication. Parts of the text are reused here:

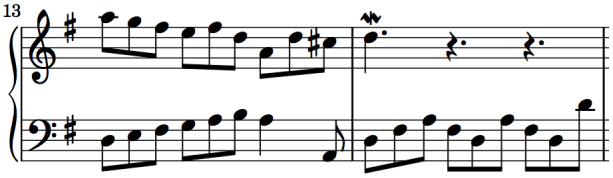
- M. Dorfer, J. j. Hajič, A. Arzt, H. Frostel, and G. Widmer. Learning audio—sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval*, 1 (1):22–33, 2018a. doi:http://doi.org/10.5334/tismir.12.

Personal Contributions I am responsible for designing the structure of the dataset and implemented the basis of the MSMD code base. Jan later on extended my "sheet-manager" project, a tool I wrote for the semi-automatic annotation of sheet music images, to automatically process and annotate the data of MSMD.

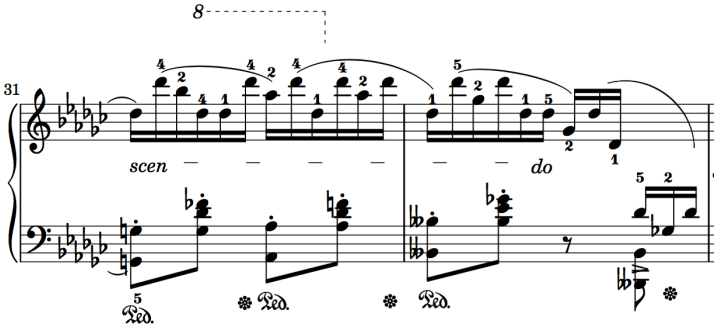
2.1 Description of Dataset

The dataset is based on the Mutopia collection of LilyPond-encoded[2] pieces and is created entirely automatically from the Mutopia project repository.[3] Some exam-

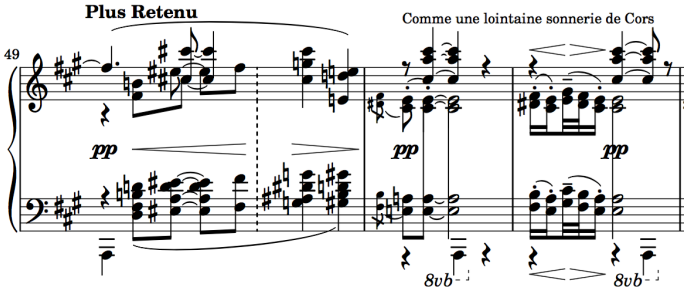
[1] `http://www.mutopiaproject.org`
[2] `http://www.lilypond.org`
[3] `https://github.com/MutopiaProject/MutopiaProject`, commit code `e325d7`



Simple: J. S. Bach, Invention in G



Medium: F. Chopin, Etude op. 10 no. 5



Complex: C. Debussy, Prélude IV, L.117

Figure 2.1: Example scores illustrating the range of music in MSMD, from simple to complex.

ples illustrating the variety of music in MSMD are shown in Figure 2.1. MSMD contains 479 solo piano pieces of mostly classical music by 53 composers, for a total of 1,129 pages of music and more than 15 hours of aligned audio. The pieces are available in two modalities: as scores (sheet music), and as MIDI, both exported directly from LilyPond. We extract staff and notehead locations and pitches from the score, and synthesize audio from the MIDI file and compute spectrograms (see Section 2.1.2 below). Then, we align the three modalities — noteheads in score;

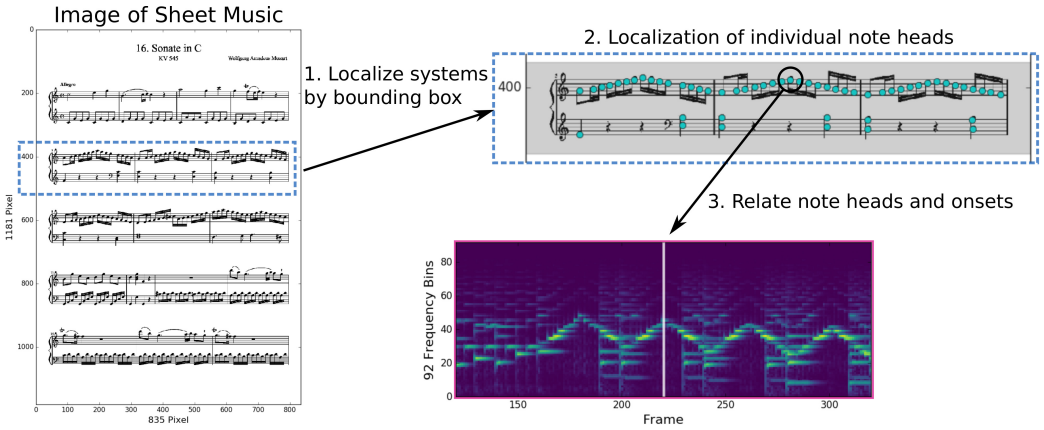


Figure 2.2: Core dataset workflow. For producing the alignment, it is necessary to "unroll" the score using individual staff systems, so that the ordering of noteheads in the score corresponds to the ordering of the notes in the MIDI file. The pixel position (coordinates) of each individual notehead are then linked to their respective counterparts in the audio (and MIDI).

MIDI events; audio/spectrogram timeline — using temporal and pitch information that is provided by the Lilypond-MIDI connection. Figure 2.2 summarizes how this alignment relates the modalities at hand.

What makes the dataset valuable, and makes the experiments described in this thesis possible, is that the modalities are automatically aligned at a fine-grained level: each individual notehead in the scores is linked to its counterpart MIDI event(s) in the audio modality. There is a total of 344,742 such aligned pairs. We will see in the remaining chapters of Part I that this is exactly the kind of annotation required to train a wide range of different machine learning models. In particular, I will show how to utilize three different machine learning paradigms all building on top of precisely annotated and aligned audio–sheet music data.

2.1.1 Recommended Train/Test Splits

We consider three scenarios that motivate how MSMD should be split into a training, validation and a test set. First, we consider simply a random mix of all the available pieces, denoted in the experimental results as *all* (see result sections in Chapter 4 and Chapter 5).

Next, for experiments that are to focus on a stylistically homogeneous body of music, we suggest using only the works of a single composer. In the case of MSMD, this would be Johann Sebastian Bach, since there are enough of his works in MSMD to allow training on this set, and their style is consistent. Experiments with this

	part	all	bach-only	bach-out
# Pieces	train	360	100	281
	valid	19	23	25
	test	100	50	173
# Pages	train	970	251	725
	valid	28	40	25
	test	131	88	379
# Noteheads	train	316,038	77,834	235,590
	valid	6,907	10,805	4,834
	test	29,851	23,733	112,372
# Events	train	310,377	75,283	233,041
	valid	6,583	10,379	4,772
	test	29,811	23,296	108,958
# Aln. Pairs	train	308,761	74,769	231,617
	valid	6,660	10,428	4,809
	test	29,321	23,119	108,316

Table 2.1: MSMD statistics for the recommended train/test splits. Note that the numbers of noteheads, events, and aligned pairs do not match. This is because (a) not every notehead is supposed to be played, esp. tied notes; (b) some onsets do not get a notehead of their own, e.g. ornaments; (c) sometimes the alignment algorithm makes mistakes.

train/test split are labeled *bach-only*.

Finally, for specialized experiments targeted at the generalization to a previously unseen musical style, we propose to leave one composer out of the training/validation data (again, J.S. Bach) and use his pieces only for testing. Experiments with this train/test split are labeled *bach-out*.

The exact piece lists defining the splits, including the split of the training data into train and validation sets, are included with the dataset. The statistics for the splits are given in Table 2.1.

2.1.2 Spectrogram Computation

To make it easier to start working with the MSMD dataset we deliver it with a set of preprocessed audio recordings. In particular, we compute log-frequency spectrograms of the audios, with a sample rate of 22.05kHz and an FFT window size of 2048 samples. For dimensionality reduction we apply a normalized logarithmic filterbank with 16-bands per octave, allowing only frequencies from 30Hz to 6kHz. This results in 92 frequency bins. The frame rate of the spectrogram is 20 frames

per second.

We generate up to 7 performances per piece, using various tempo ratios between 0.9 and 1.1 of the original MIDI tempo and four open-source piano soundfonts. One of the four soundfonts is reserved for testing (it is never used for spectrograms seen in training). Providing different audio versions of one piece can bee seen as a task specific form of data augmentation to synthetically increase the dataset and to learn models which are then robust against these variations. The experiments in Chapter 4 will show that this is indeed the case.

I want to emphasize again, that this is just a suggestion and a support to reduce initial hurdles for working with MSMD. If required, it is easily possible to prepare different or additional versions of the audio as well as the spectrograms.

2.1.3 Potential Applications

Finally, we see a large number of potential applications of the MSMD dataset introduced in this chapter. Recall that the dataset comes with a rich set of annotations and alignments. In particular, we know for each note head in each sheet image its pixel position as well as its corresponding MIDI note-event and therefore also its onset time, pitch and duration in the synthesized audios. Consequently, we expect that MSMD will become a valuable resource for future work on topics such as:

- Optical Music Recognition (OMR)
- Off-line Alignment of Sheet Images to Audio
- (Real-time) Score-Following in Sheet Images
- Sheet-Informed Transcription (i.e., to detect errors in a performance while practicing)
- Piece and Performance Retrieval as a Service for Musicians.

We hope that the research community will make use of this dataset, which we believe brings many sheet music related MIR tasks in reach of state-of-the-art machine learning methods.

2.2 Dataset Details

For the reader who is perhaps intrigued by the options offered in Section 2.1.2, we provide a more thorough description of the MSMD dataset and the process through which it was built, so that its strength and limitations are more transparent. The overall structure and workflow of MSMD is captured in Figure 2.3. For those

readers who are not interested in the very details of how the dataset is constructed I recommend to proceed to the first methodological proposal presented in Chapter 3.

MSMD is structured into *pieces*. Pieces are abstract musical entities, encoded with a LilyPond file extracted from Mutopia, that can be embodied in MSMD either as *scores*, the visual modality, or *performances*, the audio modality. We extract various *views* of a score, and *features* of a performance. Finally, we align noteheads in the score to note events in the performances. The information available for each piece and how it is derived is visualized in Figure 2.3.

MSMD contains 479 solo piano pieces of mostly classical music by 53 composers, with a total of 344,742 aligned notehead/note event pairs for training fine-grained multimodal models. More detailed statistics are given in Table 2.2. Both the audio and visual modality is synthesized.

2.2.1 Extraction from Mutopia

MSMD is built from the Mutopia open-source collection,[4] which contains 2099 pieces by 317 composers, encoded in the LilyPond representation. The collection has an open license, it is large enough for machine learning experiments, and, crucially, the PDF file generated by LilyPond contains additional data that eliminates the need for OMR in data preparation.

However, Mutopia has few editorial guidelines for contributors. It is even not immediately clear which files constitute a single piece. Mutopia contributors often make use of LilyPond's `\include` mechanism, to the extent that some files are shared between multiple pieces. Therefore, in order to extract the pieces, preprocessing was necessary: expanding LilyPond's `\include` directives to merge individual source files to one, determining which file is the root file for the given piece, discovering its header to select piano pieces, and some standardization, e.g. against unwrapping repeats in MIDI.

LilyPond itself enforces no consistency in encoding practices and no DOM parser is available for querying the document. Therefore, this preprocessing was implemented *ad hoc*, dealing with the raw LilyPond file, except for using the Abjad package[5] to convert LilyPond substrings into MIDI pitch codes.

2.2.2 Building MSMD

We generated MSMD only from solo piano (and harpsichord or clavichord) pieces, which are nevertheless the musically most complex group. After extracting relevant pieces from Mutopia, we synthesized their performances and scores, and finally

[4] `https://github.com/MutopiaProject/MutopiaProject`, commit code `e325d7`
[5] `http://abjad.mbrsi.org`

aligned the modalities at the individual note level, which enables us to train cross-modal models. The pipeline is fully automatic; overall, it was successful in 479 out of 697 of the piano pieces in Mutopia (69 %).

We first normalize the input LilyPond file (e.g., compiler version or language of pitch names). with `convert-ly`; pitches are converted to absolute, and pitch names are set to English. LilyPond then generates both a PDF and the MIDI of the piece. We then build the score and performances from these files (see Figure 2.3) and align them.

2.2.2.1 Scores

In the visual domain, we use the default PDF file generated from LilyPond; each piece in MSMD only has one score. Three views are generated:

- Images: pages of the PDF are exported as images with a pre-defined width (835);
- Coords: the coordinates of noteheads and staff systems are extracted per page;
- MuNG – MUSCIMA++ Notation Graph (Hajič jr and Pecina, 2017): records locations of the noteheads and systems, which noteheads correspond to which system, and cross-modality alignment.

The MuNG (MUSCIMA++ Notation Graph) format explicitly records how noteheads are grouped by systems, and it has facilities for recording the alignment between the visual elements and their counterparts in the performances. Furthermore, it ensures future interoperability with OMR tools.

First, we detect noteheads. Instead of OMR, we exploit the *point-and-click* feature of LilyPond: it adds for each notehead a cross-reference into the PDF with its bounding box, and a pointer to its origin in the LilyPond file,[6] from which we can parse its pitch.

We then (1) detect system regions, so that we can then properly "unroll" the score, and (2) assign noteheads to the appropriate systems, so that we can group noteheads into simultaneities when aligning the score to the performance. We binarize the image and apply morphological opening using a structuring element with a height of 1 px and width of 0.6 of the image. As we only use piano music, the bounding boxes of groups of 10 foreground components, top-down, can then be considered system regions.

To assign noteheads to systems, we apply connected component search to the binarized page and group all noteheads that are part of a connected component

[6] We mine this from the PDF with the *pdfminer* package.

overlapping a system region; Noteheads in components that do not overlap a system region are added to the system of the vertically closest component that does overlap one. This is not perfect: if two systems are connected, e.g. because a slur of one intersects a beam of the other, they form a single component, noteheads cannot be correctly grouped. Fortunately, system detection mistakes happen rarely, and the affected pieces are subsequently discarded, since this introduces mistakes into the alignment (see below).

2.2.2.2 Performances

For each performance, we first generate a *performance MIDI* by scaling the tempo of the piece MIDI between 0.9 and 1.1, and we render the performance audio with fluidsynth[7] and a piano soundfont. Then, *performance features* are extracted, using the `madmom` library:[8]

- Spectrogram (for details, see Section 2.1.2),
- Note events list: a note event is a quituplet *(onset, pitch, duration, velocity, channel)* extracted from the performance MIDI by pairing corresponding note-on and note-off events,
- Onsets list: synchronizes MIDI onset times with spectrogram frames,
- MIDI matrix: rows are pitches, columns are frames, a cell is 1 if the pitch is active in the frame.

For all frame-wise features (spectrogram, onsets and MIDI matrix), we set the rate to 20 FPS. Beyond extracting the spectrogram, the audio itself is not needed. In the version of MSMD used for the experiments in this thesis, we therefore discard it to conserve storage space; it can of course be kept if needed.

We generate up to 7 performances per piece, using various tempo ratios and soundfonts as a form of audio data augmentation.

2.2.2.3 Alignment

Now that the audio and score modalities have been generated, we perform fine-grained alignment, which is the main "added value" of MSMD that allows training multimodal models such as the ones presented in this thesis.

To each *notehead* in the score, we assign its corresponding *note event*, if one exists (e.g., noteheads on the right-hand side of a tie have no onset by definition). We use Dynamic Time Warping (DTW) applied to *simultaneities*, sets of objects that share

[7] http://www.fluidsynth.org/
[8] https://github.com/CPJKU/madmom

a point in musical time – essentially, single notes or chords. A simultaneity in a performance is a set of note events that has the same onset time. A simultaneity in a score is a set of noteheads that (1) are arranged vertically and (2) are associated with the same system (this is why explicitly assigning noteheads to systems was necessary).

We have pitch information for both modalities at the target granularity level. From the performance MIDI, we get this directly for each note event; for noteheads in the score, we extract pitch by parsing the corresponding LilyPond token to which the point-and-click backlink from the PDF leads.

We then use the Dice coefficient of their pitch sets of a given pair of simultaneities as the DTW inverse cost function. A second round of DTW aligns the elements *within* the matched simultaneities, unrolled bottom to top by pitch and using pitch equivalence as the inverse cost function. We only retain alignment for element pairs with matching pitches.

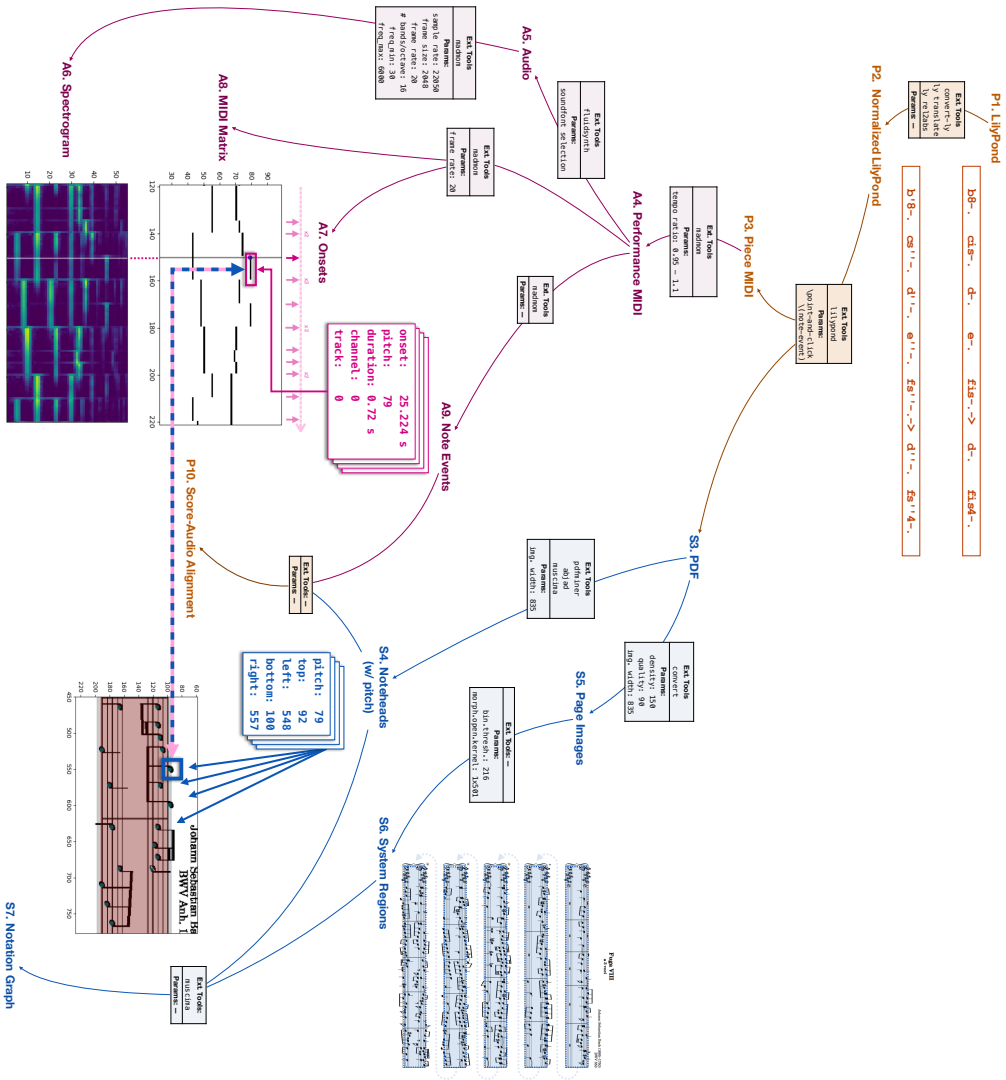
Because there is only one score and the performances are synthesized with no insertions, deletions, or synchronization changes, it is sufficient to align only one score/performance pair for each piece.

If more than 5 % of the noteheads on any page are not paired with a note event with a matching pitch, we discard the piece from MSMD.[9]

A visualization of an aligned notehead/note event pair and its surroundings in both modalities is given in Figure 2.4.

[9] The largest source of failures by far was the “q“ token of LilyPond, which means "repeat last simultaneity". This is very practical from the engraver’s point of view, but near-impossible to resolve without either manipulating LilyPond’s rendering engine, or having a DOM parser.

Figure 2.3: The fully automated workflow for building an MSMD piece, given its LilyPond file. External tools and configuration values are listed for each individual build step. We visualize some of the generated elements and their relationships to illustrate how they fit into this "big picture"; a more detailed view of their relationships is in Figure 2.4.



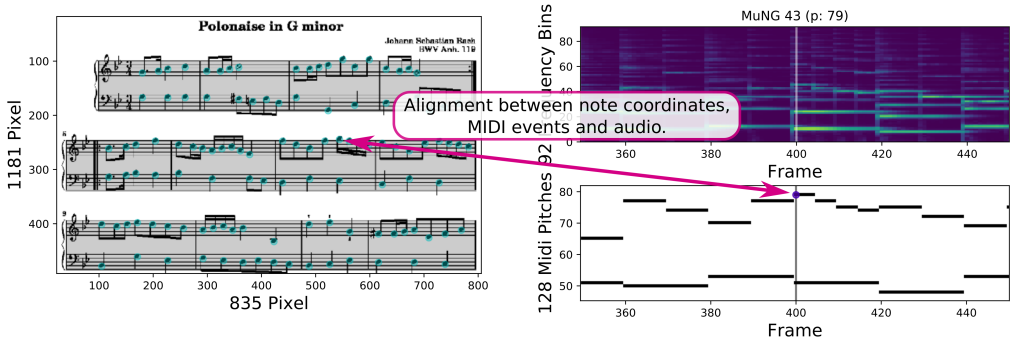


Figure 2.4: An aligned notehead/note event pair across the modalities, with context. For the visual modality, we show the score itself: notehead coordinates are the turquoise dots, the system region is given in gray. In the two panels on the right side, we show how the note is aligned to the spectrogram and to the MIDI matrix (the vertical line signifies its frame; the purple dot in the MIDI matrix signifies that the alignment is correct).

Composer	Pieces	Pgs	(Avg)	Audio mins.	(Avg)	Notes	(Avg.)	MIDI Events	(Avg.)	Aligned Pairs	(Avg.)
BachJS	173	379	2.19	297.18	1.72	112372	649.55	108958	629.82	**108316**	626.10
BeethovenLv	29	171	5.90	108.30	3.73	59877	2064.72	59365	2047.07	**59206**	2041.59
ChopinFF	16	59	3.69	33.61	2.10	18054	1128.38	17815	1113.44	**17798**	1112.38
MozartWA	19	57	3.00	39.15	2.06	16213	853.32	16489	867.84	**16023**	843.32
SousaJP	8	30	3.75	15.93	1.99	12268	1533.50	12058	1507.25	**12075**	1509.38
SchumannR	25	44	1.76	32.38	1.30	11661	466.44	11140	445.60	**11437**	457.48
JoplinS	7	26	3.71	20.11	2.87	11781	1683.00	11054	1579.14	**11158**	1594.00
Traditional	57	58	1.02	25.92	0.45	10520	184.56	11295	198.16	**10492**	184.07
VerdiG	6	23	3.83	29.55	4.92	10080	1680.00	10007	1667.83	**9998**	1666.33
MendelsBarF	6	20	3.33	23.46	3.91	7833	1305.50	7675	1279.17	**7718**	1286.33
HandelGF	20	25	1.25	17.98	0.90	7042	352.10	6996	349.80	**7014**	350.70
TchaikovskyPI	5	20	4.00	16.64	3.33	6939	1387.80	6560	1312.00	**6759**	1351.80
Rimsky-KorN	5	17	3.40	13.51	2.70	6237	1247.40	5625	1125.00	**6154**	1230.80
MussorgskyM	7	19	2.71	14.10	2.01	6444	920.57	6271	895.86	**6075**	867.86
BurgmullerJFF	17	20	1.18	11.17	0.66	6011	353.59	5876	345.65	**5948**	349.88
HaydnFJ	5	13	2.60	13.49	2.70	5522	1104.40	5451	1090.20	**5453**	1090.60
SchubertF	2	14	7.00	13.00	6.50	4467	2233.50	5464	2732.00	**4363**	2181.50
KumarR	4	11	2.75	18.66	4.66	3230	807.50	3182	795.50	**3190**	797.50
CzernyC	17	17	1.00	6.08	0.36	2726	160.35	2688	158.12	**2725**	160.29
StraussJJ	2	5	2.50	3.92	1.96	2718	1359.00	2702	1351.00	**2683**	1341.50
SatieE	5	14	2.80	17.77	3.56	2579	515.80	2515	503.00	**2531**	506.20
GriegE	1	6	6.00	3.25	3.25	2196	2196.00	2184	2184.00	**2187**	2187.00
ScriabinA	4	9	2.25	6.73	1.68	2045	511.25	1844	461.00	**2001**	500.25
Other	*39*	*72*	*1.85*	*72.92*	*1.87*	*23981*	*614.90*	*23557*	*604.03*	**23438**	*600.97*
Total	479	1129	2.36	854.81	1.78	352796	736.53	346771	723.95	**344742**	719.71

Table 2.2: Detailed statistics on the MSMD dataset. We give the numbers of pieces, pages, audio length, notes in the score, MIDI note events, and the total number of aligned notehead-MIDI note pairs, and the per-piece averages of those numbers, for individual composers in the dataset. Composers with less than 2000 aligned events are aggregated in the *Other* category; these usually only contribute one or two short pieces.

3 Supervised Function Approximation for Score Following in Sheet Music

In this chapter, I describe the first out of three machine learning paradigms utilized to learn task-specific representations from annotated audio – sheet music data. In particular, this chapter addresses the matching of short music audio snippets to their corresponding pixel locations in images of sheet music via supervised function approximation. I propose a system that simultaneously learns to read notes, "listens" to the currently playing music, and matches the music to its corresponding notes in the sheet. The system is built around an end-to-end multimodal convolutional neural network that takes as input images of sheet music and spectrograms of short snippets of audio. I already note at this point that the general design of this network architecture will be reused in the methods proposed in Chapters 4 and 5. The network then learns to predict, for a given unseen audio snippet (covering approximately one bar of music), the corresponding position in the respective score line. The experimental results provide first empirical evidence that with the use of (deep) convolutional neural networks – which have already proven to be powerful image processing models in other domains (Krizhevsky et al., 2012) – working with sheet music becomes feasible and a promising future research direction. We will see in the remaining chapters of Part I that this is indeed the case. This chapter is based on the following publication:

- M. Dorfer, A. Arzt, and G. Widmer. Towards score following in sheet music images. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 789–795, New York City, USA, 2016b.

Personal Contributions I am responsible for developing and implementing the method and carried out all the experiments.

3.1 Introduction

Precisely linking a performance to its respective sheet music – commonly referred to as audio-to-score alignment – is an important topic in MIR and the basis for many applications (Thomas et al., 2012). For instance, the combination of score

and audio supports algorithms and tools that help musicologists in in-depth performance analysis (see e.g. (Cook, 2007)), allows for new ways to browse and listen to classical music (e.g. (Dunn et al., 2006; Melenhorst et al., 2015)), and can generally be helpful in the creation of training data for tasks like beat tracking or chord recognition. When done on-line, the alignment task is known as score following, and enables a range of applications like the synchronization of visualizations to the live music during concerts (e.g. (Arzt et al., 2015; Prockup et al., 2013)), and automatic accompaniment and interaction live on stage (e.g. (Cont, 2009; Raphael, 2010)).

So far all approaches to this task depend on a symbolic, computer-readable representation of the sheet music, such as MusicXML or MIDI (see e.g. (Arzt et al., 2015; Prockup et al., 2013; Cont, 2009; Raphael, 2010; Müller et al., 2005; Niedermayer and Widmer, 2010; Miron et al., 2014; Duan and Pardo, 2011; Izmirli and Sharma, 2012)). This representation is created either manually (e.g. via the time-consuming process of (re-)setting the score in a music notation program), or automatically via optical music recognition software. Unfortunately automatic methods are still unreliable and thus of limited use, especially for more complex music like orchestral scores (Thomas et al., 2012).

The central idea presented in this chapter is to develop a method that links the audio and the image of the sheet music *directly*, by *learning* correspondences between these two modalities, and thus making the complicated step of creating an in-between representation obsolete. We aim for an algorithm that simultaneously learns to *read notes*, *listens* to music and *matches* the currently played music with the correct notes in the sheet music. We will tackle the problem in an end-to-end neural network fashion, meaning that the entire behaviour of the algorithm is learned purely from data and no further manual feature engineering is required.

3.2 Methods

This section describes the audio-to-sheet matching model and the input data required, and shows how the model is used at test time to predict the expected location of a new unseen audio snippets in the respective sheet image.

3.2.1 Data, Notation and Task Description

The model takes two different input modalities at the same time: images of scores, and short excerpts from spectrograms of audio renditions of the score (we will call these *query snippets* as the task is to predict the position in the score that corresponds to such an audio snippet). For this first proof-of-concept, we make a number of simplifying assumptions: for the time being, the system is fed only a *single staff line* at a time (not a full page of score). We restrict ourselves to

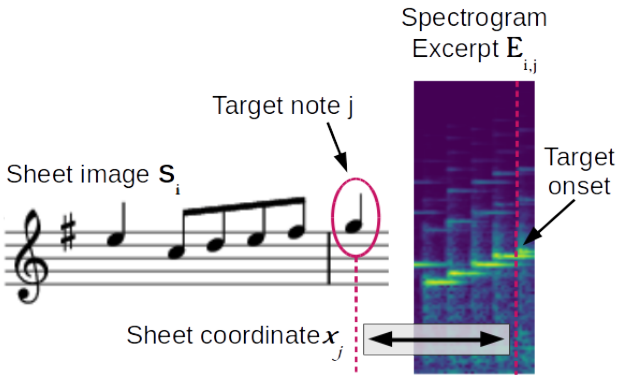


Figure 3.1: Spectrogram-to-sheet correspondence. In this example the rightmost onset in spectrogram excerpt $\mathbf{E}_{i,j}$ corresponds to the rightmost note (target note j) in sheet image $\mathbf{S}_i$. For the present case the temporal context of about 1.2 seconds (into the past) covers five additional notes in the spectrogram. The staff image and spectrogram excerpt are exactly the multimodal input presented to the proposed audio-to-sheet matching network. At train time the target pixel location x_j in the sheet image is available; at test time $\hat{x}_j$ has to be predicted by the model (see Figure 3.2).

monophonic music, and to the *piano*. To generate training examples, we produce a fixed-length query snippet for each note (onset) in the audio. The snippet covers the target note onset plus a few additional frames, at the end of the snippet, and a fixed-size context of 1.2 seconds into the past, to give some temporal context. The same procedure is followed when producing example queries for off-line testing.

A training/testing example is thus composed of two inputs: Input 1 is an image $\mathbf{S}_i$ showing one staff of sheet music. Input 2 is an audio snippet – specifically, a spectrogram excerpt $\mathbf{E}_{i,j}$ – cut from a recording of the piece, of fixed length. The rightmost onset in spectrogram excerpt $\mathbf{E}_{i,j}$ is interpreted as the target note j whose position we want to predict in staff image $\mathbf{S}_i$. For the music used in our experiments (Section 3.3) this context is roughly one bar of music. For each note j (represented by its corresponding spectrogram excerpt $\mathbf{E}_{i,j}$) we are given its *ground truth* sheet location x_j in sheet image $\mathbf{S}_i$ by annotation. Coordinate x_j is the distance of the note head (in pixels) from the left border of the image. As we work with unrolled systems of sheet music we only need the x-coordinate of the note at this point. Figure 3.1 relates all components involved.

Summary and Task Description: For training we present triples of (1) staff image $\mathbf{S}_i$, (2) spectrogram excerpt $\mathbf{E}_{i,j}$ and (3) ground truth pixel x-coordinate x_j to our audio-to-sheet matching model. At test time only the staff image and spectrogram

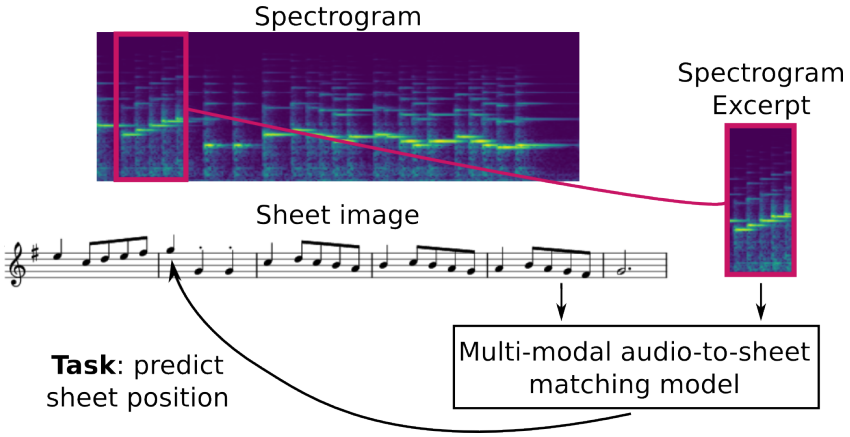


Figure 3.2: Schematic sketch of the audio-to-sheet matching task. Given a sheet image $\mathbf{S}_i$ and a short snippet of audio (spectrogram excerpt $\mathbf{E}_{i,j}$) the model has to predict the audio snippet's corresponding pixel location x_j in the image.

excerpt are available and the task of the model is to predict the estimated pixel location $\hat{x}_j$ in the image. Figure 3.2 shows a sketch summarizing this task.

3.2.2 Audio-Sheet Matching as Bucket Classification

We now propose a multimodal convolutional neural network architecture that learns to match unseen audio snippets (spectrogram excerpts) to their corresponding pixel location in the sheet image.

3.2.2.1 Network Structure

Figure 3.3 provides a general overview of the deep network and the proposed solution to the matching problem. As mentioned above, the model operates jointly on a staff image $\mathbf{S}_i$ and the audio (spectrogram) excerpt $\mathbf{E}_{i,j}$ related to a note j. The rightmost onset in the spectrogram excerpt is the one related to target note j. The multimodal model consists of two specialized convolutional networks: one dealing with the sheet image and one dealing with the audio (spectrogram) input. In the subsequent layers we fuse the specialized sub-networks by concatenation of the latent image- and audio representations and additional processing by a sequence of dense layers. For a detailed description of the individual layers we refer to Table 3.1 in Section 3.3.4. The output layer of the network and the corresponding localization principle are explained in the following.

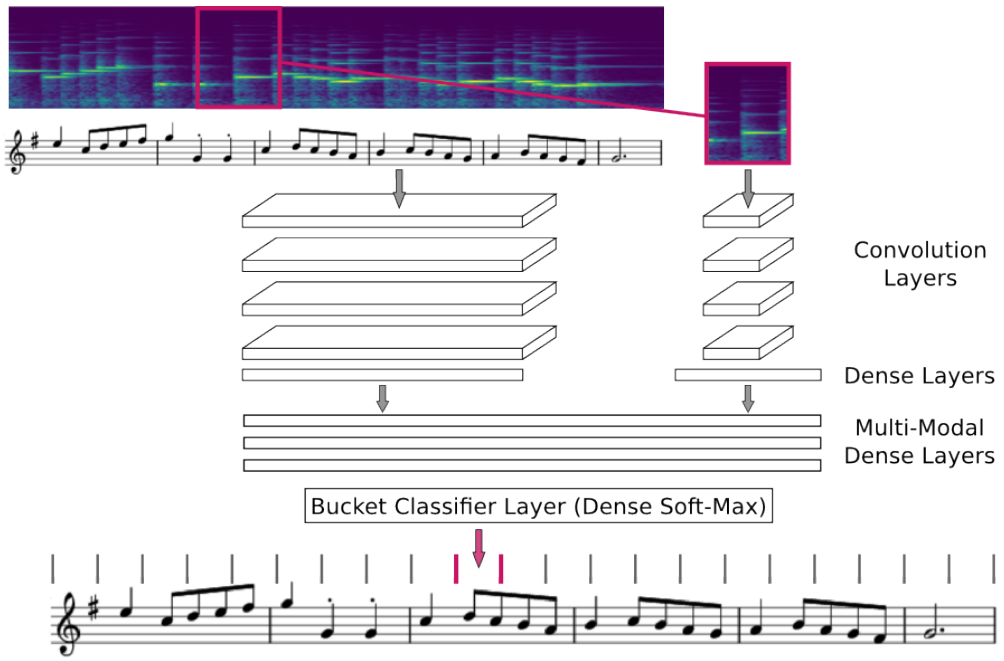


Figure 3.3: Overview of multimodal convolutional neural network for audio-to-sheet matching. The network takes a staff image and a spectrogram excerpt as input. Two specialized convolutional network parts, one for the sheet image and one for the audio input, are merged into one multimodality network. The output part of the network predicts the region in the sheet image – the classification bucket – to which the audio snippet corresponds.

3.2.2.2 Audio-to-Sheet Bucket Classification

The objective for an unseen spectrogram excerpt and a corresponding staff of sheet music is to predict the excerpt's location x_j in the staff image. For this purpose we start with horizontally quantizing the sheet image into B non-overlapping buckets. This discretisation step is indicated as the short vertical lines in the staff image above the score in Figure 3.3. In a second step we create for each note j in the train set a target vector $\mathbf{t}_j = \{t_{j,b}\}$ where each vector element $t_{j,b}$ holds the probability that bucket b covers the current target note j. In particular, we use soft targets, meaning that the probability for one note is shared between the two buckets closest to the note's true pixel location x_j. We linearly interpolate the shared probabilities based on the two pixel distances (normalized to sum up to one) of the note's location x_j to the respective (closest) bucket centers. Bucket centers are denoted by c_b in the following where subscript b is the index of the respective bucket. Figure 3.4 shows an example sketch of the components described above. Based on the soft

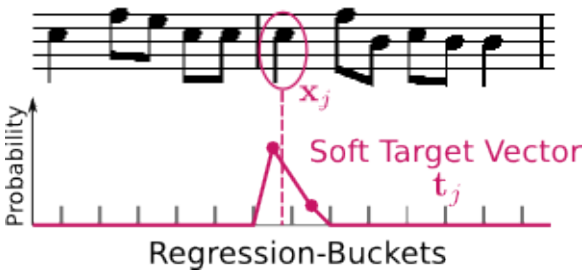


Figure 3.4: Part of a staff of sheet music along with soft target vector $\mathbf{t}_j$ for target note j surrounded with an ellipse. The two buckets closest to the note share the probability (indicated as dots) of containing the note. The short vertical lines highlight the bucket borders.

target vectors we design the output layer of our audio-to-sheet matching network as a B-way soft-max with activations defined as:

$$\phi(y_{j,b}) = \frac{e^{y_{j,b}}}{\sum_{k=1}^{B} e^{y_{j,k}}} \tag{3.1}$$

$\phi(y_{j,b})$ is the soft-max activation of the output neuron representing bucket b and hence also representing the region in the sheet image covered by this bucket. By applying the soft-max activation the network output gets normalized to range $(0, 1)$ and further sums up to 1.0 over all B output neurons. The network output can now also be interpreted as a vector of location probabilities $\mathbf{p}_j = \{\phi(y_{j,b})\}$ and shares the same value range and properties as the soft target vectors.

In training, we optimize the network parameters Θ by minimizing the Categorical Cross Entropy (CCE) loss l_j between target vectors $\mathbf{t}_j$ and network output $\mathbf{p}_j$:

$$l_j(\Theta) = -\sum_{k=1}^{B} t_{j,k}\, log(p_{j,k}) \tag{3.2}$$

The CCE loss function becomes minimal when the network output $\mathbf{p}_j$ exactly matches the respective soft target vector $\mathbf{t}_j$. In Section 3.3.4 we provide further information on the exact optimization strategy used.[1]

[1] *For the sake of completeness*: In our initial experiments we started to predict the sheet location of audio snippets by minimizing the Mean-Squared-Error (MSE) between the predicted and the true pixel coordinate (MSE regression). However, we observed that training these networks is much harder and further performs worse than the bucket classification approach proposed in this chapter.

3.2.3 Sheet Location Prediction

Once the model is trained, we use it at test time to predict the expected location $\hat{x}_j$ of an audio snippet with target note j in a corresponding image of sheet music. The output of the network is a vector $\mathbf{p}_j = \{p_{j,b}\}$ holding the probabilities that the given test snippet j matches with bucket b in the sheet image. Having these probabilities we consider two different types of predictions: (1) We compute the center c_b^* of bucket $b^* = \text{argmax}_b\, p_{j,b}$ holding the highest overall matching probability. (2) For the second case we take, in addition to b^*, the two neighbouring buckets $b^* - 1$ and $b^* + 1$ into account and compute a (linearly) probability weighted position prediction in the sheet image as

$$\hat{x}_j = \sum_{k \in \{b^*-1, b^*, b^*+1\}} w_k c_k \tag{3.3}$$

where weight vector $\mathbf{w}$ contains the probabilities $\{p_{j,b^*-1}, p_{j,b^*}, p_{j,b^*+1}\}$ normalized to sum up to one and c_k are the center coordinates of the respective buckets.

3.3 Experimental Evaluation

This section evaluates our audio-to-sheet matching model on a publicly available dataset. We describe the experimental setup, including the data and evaluation measures, the particular network architecture as well as the optimization strategy, and provide quantitative results.

3.3.1 Experiment Description

The aim of this chapter is to show that it is feasible to learn correspondences between audio (spectrograms) and images of sheet music in an *end-to-end* neural network fashion, meaning that an algorithm learns the entire task purely from data, so that no hand crafted feature engineering is required. We try to keep the experimental setup simple and consider one staff of sheet music per train/test sample (this is exactly the setup drafted in Figure 3.3). To be perfectly clear, the task at hand is the following: For a given audio snippet, find its x-coordinate pixel position in a corresponding staff of sheet music. For now, we further restrict the audio to monophonic music containing half, quarter and eighth notes but allow variations such as dotted notes, notes tied across bar lines as well as accidental signs.

Note, that we will re-evaluate the approach at hand using the more complex piano music of MSMD (Chapter 2) to compare it to a related, reinforcement learning based method, in Chapter 5.

3.3.2 Data

For a first evaluation of our approach we consider the Nottingham[2] data set which was used, e.g., for piano transcription in (Boulanger-lewandowski et al., 2012). It is a collection of midi files already split into train, validation and test tracks. To be suitable for audio-to-sheet matching we prepare the data set (midi files) as follows:

1. We select the first track of the midi files (right hand, piano) and render it as sheet music using Lilypond.[3]
2. We annotate the sheet coordinate x_j of each note.
3. We synthesize the midi-tracks to *flac*-audio using Fluidsynth[4] and a *Steinway* piano sound font.
4. We extract the audio timestamps of all note onsets.

As a last preprocessing step we compute *log-spectrograms* of the synthesized flac files (Böck et al., 2016), with an audio sample rate of 22.05kHz, FFT window size of 2048 samples, and computation rate of 31.25 frames per second. For dimensionality reduction we apply a normalized 24-band logarithmic filterbank allowing only frequencies from 80Hz to 8kHz. This results in 136 frequency bins.

We already showed a spectrogram-to-sheet annotation example in Figure 3.1. In our experiment we use spectrogram excerpts covering 1.2 seconds of audio (40 frames). This context is kept the same for training and testing. Again, annotations are aligned in a way so that the rightmost onset in a spectrogram excerpt corresponds to the pixel position of target note j in the sheet image. In addition, the spectrogram is shifted 5 frames to the right to also contain some information on the current target note's onset and pitch. We chose this annotation variant with the rightmost onset as it allows for an online application of our audio-to-sheet model (as would be required, e.g., in a score following task).

3.3.3 Evaluation Measures

To quantify the performance of our approach we consider, for each test note j, the following ground truth and prediction data: (1) The true position x_j as well as the corresponding target bucket b_j (see Figure 3.4). (2) The estimated sheet location $\hat{x}_j$ and the most likely target bucket b^* predicted by the model. Given this data we compute two types of evaluation measures.

[2] www-etud.iro.umontreal.ca/~boulanni/icml2012
[3] http://www.lilypond.org/
[4] http://www.fluidsynth.org/

Sheet-Image 40×390	Spectrogram 136×40
5×5 Conv(pad-2, stride-1-2)-64-BN-ReLu	3×3 Conv(pad-1)-64-BN-ReLu
3×3 Conv(pad-1)-64-BN-ReLu	3×3 Conv(pad-1)-64-BN-ReLu
2×2 Max-Pooling + Drop-Out(0.15)	2×2 Max-Pooling + Drop-Out(0.15)
3×3 Conv(pad-1)-128-BN-ReLu	3×3 Conv(pad-1)-96-BN-ReLu
3×3 Conv(pad-1)-128-BN-ReLu	2×2 Max-Pooling + Drop-Out(0.15)
2×2 Max-Pooling + Drop-Out(0.15)	3×3 Conv(pad-1)-96-BN-ReLu
	2×2 Max-Pooling + Drop-Out(0.15)
Dense-1024-BN-ReLu + Drop-Out(0.3)	Dense-1024-BN-ReLu + Drop-Out(0.3)
Concatenation-Layer-2048	
Dense-1024-BN-ReLu + Drop-Out(0.3)	
Dense-1024-BN-ReLu + Drop-Out(0.3)	
B-way Soft-Max Layer	

Table 3.1: Architecture of Multimodal Audio-to-Sheet Matching Model: BN: Batch Normalization, ReLu: Rectified Linear Activation Function, CCE: Categorical Cross Entropy, batch size: 100

The first – the *top-k bucket hit rate* – quantifies the ratio of notes that are classified into the correct bucket allowing a tolerance of $k-1$ buckets. For example, the *top-1 bucket hit rate* counts only those notes where the predicted bucket b^* matches exactly the note's target bucket b_j. The *top-2 bucket hit rate* allows for a tolerance of one bucket and so on. The second measure – the *normalized pixel distance* – captures the actual distance of a predicted sheet location $\hat{x}_j$ to its corresponding true position x_j. To allow for an evaluation independent of the image resolution used in our experiments we normalize the pixel errors by dividing them by the width of the sheet image as $(\hat{x}_j - x_j)/width(\mathbf{S}_i)$. This results in distance errors living in range $(-1, 1)$.

We would like to emphasise that the quantitative evaluations based on the measures introduced above are performed only at time steps where a note onset is present. At those points in time an explicit correspondence between spectrogram (onset) and sheet image (note head) is established. However, in Section 3.4 we show that a time-continuous prediction is also feasible with our model and onset detection is not required at run time.

3.3.4 Model Architecture and Optimization

Table 3.1 gives details on the model architecture used for our experiments. As shown in Figure 3.3, the model is structured into two disjoint convolutional networks where one considers the sheet image and one the spectrogram (audio) input. The convolutional parts of our model are inspired by the VGG model built from sequences of small convolution kernels (e.g. 3×3) and max-pooling layers. The

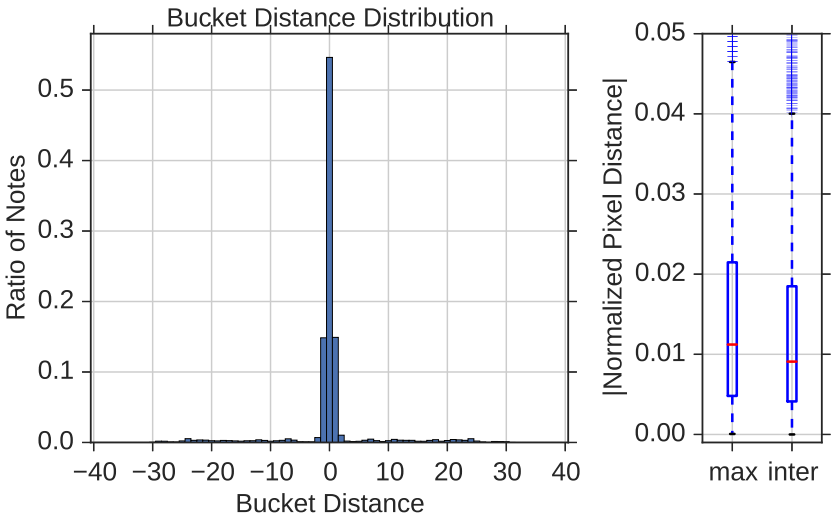


Figure 3.5: Summary of matching results on test set. *Left*: Histogram of bucket distances between predicted and true buckets. *Right*: Box-plots of absolute *normalized pixel distances* between predicted and true image position. The box-plot is shown for both location prediction methods described in Section 3.2.3 (maximum, interpolated).

central part of the model consists of a concatenation layer bringing the image and spectrogram sub-networks together. After two dense layers with 1024 units each we add a B-way soft-max output layer. Each of the B soft-max output neurons corresponds to one of the disjoint buckets which in turn represent quantised sheet image positions. In our experiments we use a fixed number of 40 buckets selected as follows: We measure the minimum distance between two subsequent notes – in our sheet renderings – and select the number of buckets such that each bucket contains at most one note. It is of course possible that no note is present in a bucket – e.g., for the buckets covering the clef at the beginning of a staff. As activations function for the inner layers we use rectified linear units (Glorot et al., 2011) and apply batch normalization (Ioffe and Szegedy, 2015) after each layer as it helps training and convergence.

Given this architecture and data we optimize the parameters of the model using mini-batch stochastic gradient descent with Nesterov style momentum (Nesterov, 1983; Sutskever et al., 2013). We set the batch size to 100 and fix the momentum at 0.9 for all epochs. The initial learn-rate is set to 0.1 and divided by 10 every 10 epochs. We additionally apply a weight decay of 0.0001 to all trainable parameters of the model.

	Train	**Valid**	**Test**
Top-1-Bucket-Hit-Rate	79.28%	51.63%	54.64%
Top-2-Bucket-Hit-Rate	94.52%	82.55%	84.36%
mean($\|NPD_{max}\|$)	0.0316	0.0684	0.0647
mean($\|NPD_{int}\|$)	0.0285	0.0670	0.0633
median($\|NPD_{max}\|$)	0.0067	0.0119	0.0112
median($\|NPD_{int}\|$)	0.0033	0.0098	0.0091
$\|NPD_{max}\| < w_b$	93.87%	76.31%	79.01%
$\|NPD_{int}\| < w_b$	94.21%	78.37%	81.18%

Table 3.2: *Top-k bucket hit rates* and *normalized pixel distances* (NPD) as described in Section 3.3.4 for train, validation and test set. We report mean and median of the absolute NPDs for both interpolated (int) and maximum (max) probability bucket prediction. The last two rows report the percentage of predictions not further away from the true pixel location than the width w_b of one bucket.

3.3.5 Experimental Results

Figure 3.5 shows a histogram of the signed bucket distances between predicted and true buckets. The plot shows that more than 54% of all unseen test notes are matched exactly with the corresponding bucket. When we allow for a tolerance of ± 1 bucket our model is able to assign over 84% of the test notes correctly. We can further observe that the prediction errors are equally distributed in both directions – meaning too early and too late in terms of audio. The results are also reported in numbers in Table 3.2, as the top-k bucket hit rates for train, validation and test set.

The box plots in the right part of Figure 3.5 summarize the absolute *normalized pixel distances (NPD)* between predicted and true locations. We see that the probability-weighted position interpolation (Section 3.2.3) helps improve the localization performance of the model. Table 3.2 again puts the results in numbers, as means and medians of the absolute NPD values. Finally, Fig. 3.2 (bottom) reports the ratio of predictions with a pixel distance smaller than the width of a single bucket.

3.4 Discussion and Real Music

This section provides a representative prediction example of our model and uses it to discuss the proposed approach. In the second part we then show a first step towards matching *real* (though still very simple) music to its corresponding sheet. By *real music* we mean audio that is not just synthesized midi, but played by a human on a piano and recorded via microphone.

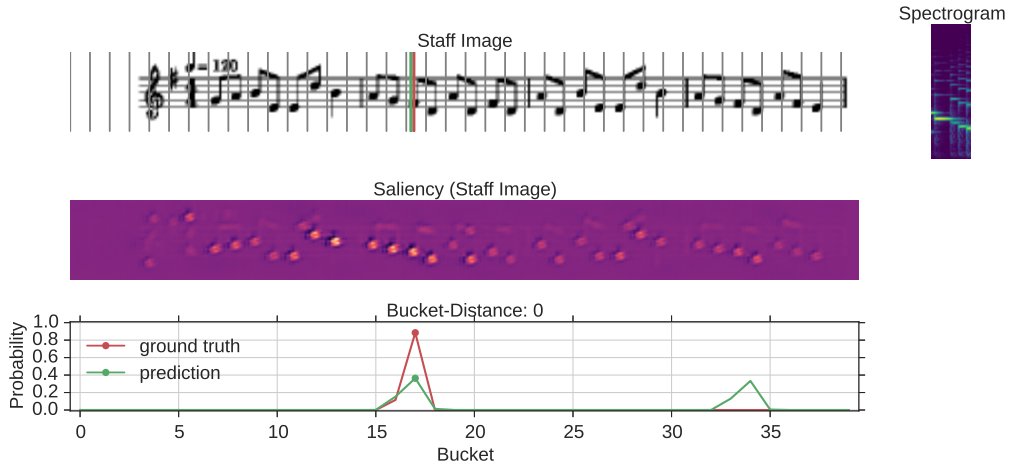


Figure 3.6: Example prediction of the proposed model. The top row shows the input staff image $\mathbf{S}_i$ along with the bucket borders as thin gray lines, and the given query audio (spectrogram) snippet $\mathbf{E}_{i,j}$. The plot in the middle visualizes the salience map (representing the attention of the neural network) computed on the input image. Note that the network's attention is actually drawn to the individual note heads. The bottom row compares the ground truth bucket probabilities with the probabilities predicted by the network. In addition, we also highlight the corresponding true and predicted pixel locations in the staff image in the top row.

3.4.1 Prediction Example and Discussion

Figure 3.6 shows the image of one staff of sheet music along with the predicted as well as the ground truth pixel location for a snippet of audio. The network correctly matches the spectrogram with the corresponding pixel location in the sheet image. However, we observe a second peak in the bucket prediction probability vector. A closer look shows that this is entirely reasonable, as the music is quite repetitive and the current target situation actually appears twice in the score. The ability of predicting probabilities for multiple positions is a desirable and important property, as repetitive structures are immanent to music. The resulting prediction ambiguities can be addressed by exploiting the temporal relations between the notes in a piece by methods such as dynamic time warping or probabilistic models. In fact, it would be possible to combine the probabilistic output of our matching model with existing score following methods, as for example (Arzt et al., 2008). In Section 3.2 we mentioned that training a sheet location prediction with MSE-regression is difficult to optimize. Besides this technical drawback it would not be straightforward to predict a variable number of locations with an MSE-model, as

the number of network outputs has to be fixed when designing the model.

In addition to the network inputs and prediction Figure 3.6 also shows a *saliency map* (Springenberg et al., 2015) computed on the input sheet image with respect to the network output.[5] The saliency can be interpreted as the input regions to which most of the net's attention is drawn. In other words, it highlights the regions that contribute most to the current output produced by the model. A nice insight of this visualization is that the network actually focuses and recognizes the heads of the individual notes. In addition it also directs some attention to the style of stems, which is necessary to distinguish for example between quarter and eighth notes.

The optimization on soft target vectors is also reflected in the predicted bucket probabilities. In particular the neighbours of the bucket with maximum activation are also active even though there is no explicit neighbourhood relation encoded in the soft-max output layer. This helps the interpolation of the true position in the image (see Figure 3.5).

3.4.2 First Steps with Real Music

As a final point, we report on first attempts at working with "real" music. For this purpose Gerhard Widmer played the right hand part of a simple piece (Minuet in G Major by Johann Sebastian Bach, BWV Anhang 114) – which, of course, was not part of the training data – on a *Yamaha AvantGrand N2* hybrid piano and recorded it using a single microphone. In this application scenario we predict the corresponding sheet locations not only at times of onsets but for a continuous audio stream (subsequent spectrogram excerpts). This can be seen as a simple version of online score following in sheet music, without taking into account the temporal relations of the predictions. We offer the reader a video[6] that shows our model following the first three staff lines of this simple piece.[7] The ratio of predicted notes having a pixel-distance smaller than the bucket width (compare Section 3.3.5) is 71.72% for this real recording. This corresponds to a average normalized-pixel-distance of 0.0402.

[5] The implementation is adopted from an example by Jan Schlüter in the recipes section of the deep learning framework *Lasagne* Dieleman et al. (2015).

[6] `https://www.dropbox.com/s/0nz540i1178hjp3/Bach_Minuet_G_Major_net4b.mp4?dl=0`

[7] Note: our model operates on single staffs of sheet music and requires a certain context of spectrogram frames for prediction (in our case 40 frames). For this reason it cannot provide a localization for the first couple of notes in the beginning of each staff at the current stage. In the video one can observe that prediction only starts when the spectrogram in the top right corner has grown to the desired size of 40 frames. We kept this behaviour for now as we see our work as a proof of concept. The issue can be easily addressed by concatenating the images of subsequent staffs in horizontal direction. In this way we will get a "continuous stream of sheet music" analogous to a spectrogram for audio.

3.5 Conclusion

In this chapter I presented a multimodal convolutional neural network which is able to match short snippets of audio with their corresponding position in the respective image of sheet music, without the need of any symbolic representation of the score. First evaluations on simple piano music suggested that this is a very promising new approach that deserves to be explored further.

As the method presented in this chapter was my first proof of concept, it naturally still has some severe limitations. So far the approach is tested only with monophonic music, notated on a single staff, and with performances that are roughly played in the same tempo as was set in our training examples.

In the remainder of Part I of this thesis we will address at least some of these limitations. In particular, I will show that the very general paradigm of applying multimodal neural networks to audio and sheet music images also scales well to the music contained in the MSMD dataset: In Chapter 4 we will start working on complex, polyphonic piano music and we will also carry out first experiments on real-world data (scanned images of sheet music and audio recordings of professional performances). In Chapter 5 we will revisit the task of score following but this time also in complex, polyphonic sheet music by utilizing a completely different machine learning paradigm.

4 Learning Cross-modal Audio – Sheet Music Embeddings

Taking the multimodal audio – sheet matching network of Chapter 3 as a basis, we will now continue with the second machine learning paradigm, which is multimodal embedding space learning. In particular, this chapter addresses the problem of matching musical audio directly to sheet music, without any higher-level abstract representation. We propose a method that learns joint embedding spaces for short excerpts of audio and their respective counterparts in sheet music images, again using multimodal convolutional neural networks. For embedding space learning we already employ the canonically correlated projection layer (CCA layer) that will be introduced and described in detail in Chapter 6 of Part II. Given the learned representations, we show how to utilize them for three sheet-music-related tasks: (1) piece/score identification from audio queries, (2) retrieving relevant performances given a score as a search query and, (3) offline audio – sheet music alignment. All retrieval models are trained and evaluated on MSMD (Chapter 2), our large scale multimodal audio – sheet music dataset, which was made publicly available along with the corresponding article (Dorfer et al., 2018a). Going beyond this synthetic training data, we carry out first retrieval experiments using scans of real sheet music of high complexity (e.g., nearly the complete solo piano works by Frederic Chopin) and commercial recordings by famous concert pianists. Our results suggest that the proposed method, in combination with the large-scale dataset, yields retrieval models that successfully generalize to data way beyond the synthetic training data used for model building.

The work presented in this chapter is based on the following publications:

- M. Dorfer and G. Widmer. Towards end-to-end audio-sheet-music retrieval. In *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, Barcelona, Spain, 2016b.
- M. Dorfer, A. Arzt, and G. Widmer. Learning audio-sheet music correspondences for score identification and offline alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 115–122, Suzhou, China, 2017a.
- M. Dorfer, J. j. Hajič, A. Arzt, H. Frostel, and G. Widmer. Learning audio—sheet music correspondences for cross-modal retrieval and piece identification.

Transactions of the International Society for Music Information Retrieval, 1 (1):22–33, 2018a. doi:http://doi.org/10.5334/tismir.12.

- M. Dorfer, J. j. Hajič, and G. Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, Kyoto, Japan, 2017b.

Personal Contributions I am responsible for developing and implementing the method and designed and carried out all the experiments.

4.1 Introduction

Many applications in Music Information Retrieval (MIR) – from retrieval scenarios to live score following to score-informed transcription – require an alignment between different representations of a piece, most often between printed score (sheet music) and recorded performance (audio). Consequently, there has been a lot of work on score-to-performance matching, with different approaches. Traditionally, automatic methods for linking audio and sheet music have relied on some common mid-level representation that allows for comparison and matching (e.g., by computation of distances or similarities) of time points in the audio and positions in the sheet music. Examples of mid-level representations are *symbolic event descriptions*, which involve the error-prone steps of automatic music transcription on the audio side (Böck and Schedl, 2012; Kelz et al., 2016; Sigtia et al., 2016; Cheng et al., 2016) and Optical Music Recognition (OMR) on the sheet music side (Wen et al., 2015; Hajič jr and Pecina, 2017; Byrd and Simonsen, 2015; Rebelo et al., 2012); or *spectral features* like pitch class profiles (chroma features), which avoid the explicit audio transcription step but still depend on variants of OMR on the sheet music side. For examples of the latter approach see, e.g., (Balke et al., 2016, 2015; Grachten et al., 2013; Kurth et al., 2007; Fremerey et al., 2009; Izmirli and Sharma, 2012).

To avoid these complications altogether, I have proposed in Chapter 3 (Dorfer et al., 2016b) the idea of directly matching sheet music images and audio, with deep neural networks. Given short excerpts of audio and the corresponding sheet music, the network learned to predict which location in the given sheet image best matches the current audio excerpt. The potential of this idea was demonstrated in the context of score following.

The approach presented in the present chapter goes beyond that in Chapter 3 in several respects. Most importantly, the original network required both sheet music and audio as input at the same time, in order to then decide which location in the sheet image best matches the current audio excerpt. We now address a more general scenario where both input modalities are required only at training time, for learning

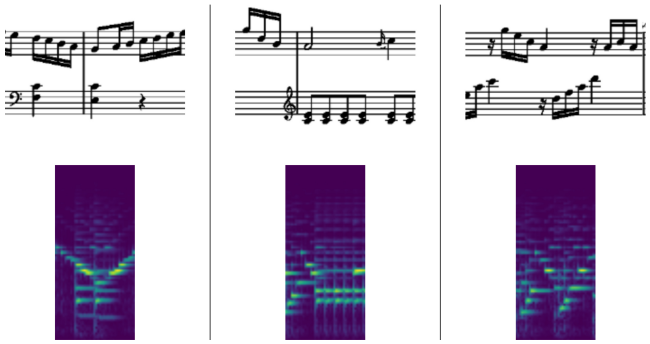


Figure 4.1: Audio –sheet music pairs presented to the network for embedding space learning.

the relation between score and audio. This requires a different network architecture that can learn two separate projections, one for embedding the sheet music and one for embedding the audio, which can then be used independently of each other. For example, we can first embed a reference collection of sheet music images using the image embedding part of the network, then embed a query audio and search for its nearest sheet music neighbours in the joint embedding space. This general scenario is referred to as *cross-modality retrieval* and supports different applications (three of which will be demonstrated in this chapter).

Specifically, we use multimodal convolutional neural networks to directly learn correspondences between images of sheet music and their respective audio counterparts. Given short excerpts of audio and corresponding sheet music images (such as the ones shown in Figure 4.1), the networks are trained to learn an embedding space in which both modalities are represented as fixed-dimensional vectors which can then be compared, e.g., via their cosine distance. To obtain a latent representation that supports this comparison, the networks employ an optimization target that encourages joint embedding spaces where semantically similar items of both modalities live close to each other.

The central idea of this approach is to circumvent the problematic definition of mid-level features by replacing it (on both sides) with a learned transformation of audio and sheet music data to a common vector space. In (Dorfer et al., 2017a) we first demonstrated how to utilize this methodology for two sheet music-related real-world applications: (1) *piece identification* via cross-modality retrieval from audio queries, and (2) *audio-to-sheet music alignment* using Dynamic Time Warping (DTW) in the learned joint embedding space. In Dorfer et al. (2018a) we further extend this work arriving at the following contributions, which we hope will greatly facilitate and accelerate future music alignment and retrieval research in the MIR community:

Contribution 1: Large Scale Audio – Sheet Music Retrieval Experiments. First experiments in (Dorfer et al., 2017a) already indicated that the approach presented in this chapter seems to scale very well with the amount of training data available, and that it is important to have as diverse a dataset as possible to arrive at a robust model. (To that end we also applied various data augmentation strategies.) Motivated by our findings in (Dorfer et al., 2017a), we prepared and published the MSMD dataset presented in Chapter 2. Given this large scale dataset, we present empirical evidence for the scalability of our approach in Section 4.3 of this chapter.

Contribution 2: Experimental Setup, Software Tools, New Experimental Baseline. To allow for an objective benchmarking of further methodological improvements we suggest a specific experimental setup on how to perform evaluations with the MSMD dataset. Additionally, to lower the initial hurdles for working with the data, we release a complete set of tools for automatically preparing, viewing, and loading the data. We present extensive experiments, using the new dataset and the suggested experimental setup. The entire experimental code including our pre-trained retrieval models is available online[1]. We hope this will serve as a basis for further research in the community.

Contribution 3: First Experiments with Substantial Real-world Data. So far, all experiments were carried out on synthetic data, e.g., rendered sheet music and audio synthesized from MIDI. In this chapter, we will report on first large-scale retrieval experiments using scanned images of real sheet music of high complexity (e.g., nearly the complete solo piano works by Frederic Chopin, from the Henle Urtext Edition) and real audio recordings by professional concert pianists. Our results suggest that the proposed method, in combination with the large-scale MSMD dataset (and appropriate data augmentation methods), yields retrieval models that successfully generalize to data way beyond the synthetic training data used for model building. This holds for sheet images and to quite some degree also for real performances.

The remainder of this chapter is structured as follows: Section 4.2 describes how to train the proposed retrieval model on the MSMD dataset, along with the applied data augmentation strategies. Sections 4.3 and 4.4 present extensive experimental results on two different retrieval tasks (sheet/audio snippet retrieval and piece/performance identification). Section 4.5 explores how the method generalizes to complex real-world data (e.g. scanned sheet music and real performances). Section 4.6 shows how to utilize the learned representation for offline audio–sheet music alignment via dynamic time warping. In Section 4.7, we summarize our results and suggest directions for further research.

[1] https://github.com/CPJKU/audio_sheet_retrieval

4.2 Learning Audio–Sheet Music Correspondences

Our approach is built around a neural network designed for learning the relationship between two different data modalities. The network learns its behavior solely from the examples presented at training time. We start this section by explaining how to prepare and post-process the MSMD dataset proposed in Chapter 2 in order to generate exactly these training examples. The final part of this section describes the underlying learning methodology in detail.

4.2.1 Data Preparation

The MSMD dataset already contains segmentations of the staff systems in all of its sheet music images (cf. Figure 2.2). In particular, we are given annotated bounding boxes around the individual systems along with the positions of the note heads associated with these systems. In addition to the scores we are also provided with audio renditions synthesized from MIDI files, or spectrograms computed from these. And most importantly, we get for each annotated note head in the image a pointer referring to its corresponding onset time in the audio. This means that we know for each notehead its location (in pixel coordinates) in the image, and its onset time in the audio. Based on this relationship and annotations, we cut out corresponding snippets of sheet music images (in our case 160×200 pixels) and short excerpts of audio represented by log-frequency spectrograms (92 bins $\times$ 42 frames, $\approx$2sec of music). Figure 4.1 shows three examples of such audio - sheet music correspondences; these are the pairs presented to our multimodal networks for training.

4.2.2 Data Augmentation

To improve the generalization ability of the resulting networks, we propose several data augmentation strategies specialized to score images and audio. In machine learning, *data augmentation* refers to the application of (realistic) data transformations in order to synthetically increase the effective size of the training set (Ronneberger et al., 2015; McFee et al., 2015). We already emphasize at this point that data augmentation is a crucial component for learning cross-modality representations that generalize to unseen music.

For **sheet image augmentation** we apply three different transformations, summarized in Figure 4.2. The first is *image scaling* where we resize the image between 95 and 105% of its original size. This should make the model robust to changes in the overall dimension of the scores. Secondly, in Δy *system translation* we slightly shift the system in the vertical direction by $\Delta y \in [-5, 5]$ pixels. We do this as the system detector will not detect each system in exactly the same way and we want our model to be invariant to such translations. In particular, it should not

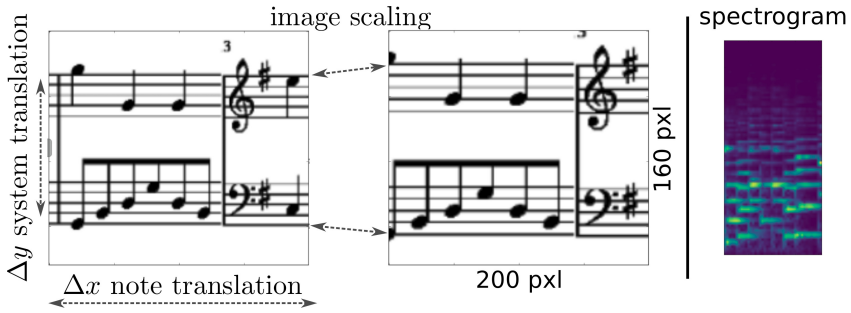


Figure 4.2: Overview of image augmentation strategies. The size of the sliding image window remains constant (160×200 pixels) but its content changes depending on the augmentations applied. The spectrogram remains the same for the augmented image versions.

be the absolute location of a note head in the image that determines its meaning (pitch) but its relative position with respect to the staff. Finally, we apply Δx *note translation*, meaning that we slightly shift the corresponding sheet image window by $\Delta x \in [-5, 5]$ pixels in the horizontal direction.

In terms of **audio augmentation**, we render the training pieces with three different sound fonts and additionally vary the tempo between 95 and 110 % of its original tempo when preparing the MSMD dataset performances (see Chapter 2). For the test set, we only use performances rendered at the original preset tempo but using an *additional unseen soundfont* (no performances using this soundfont are used for training). The test set is kept fixed to reveal the impact of the different data augmentation strategies.

4.2.3 Embedding Space Learning

This subsection describes the underlying learning methodology. As mentioned above, the core of the approach presented in this chapter is a neural network capable of learning cross-modal correspondences between short snippets of audio and sheet music images. In particular, we aim to learn a joint embedding space of the two modalities in which to perform nearest-neighbour search. One method for learning such a space, which has already proven to be effective in other domains such as text-to-image retrieval, is based on the optimization of a pairwise ranking loss (Kiros et al., 2014; Socher et al., 2014). Before explaining this optimization target, we first introduce the general architecture of our correspondence learning network. As shown in Figure 4.3 the network consists of two separate pathways f and g taking two inputs at the same time. Input one is a sheet image snippet and input two is an audio excerpt $\mathbf{A}$. This means in particular that network f is responsible for pro-

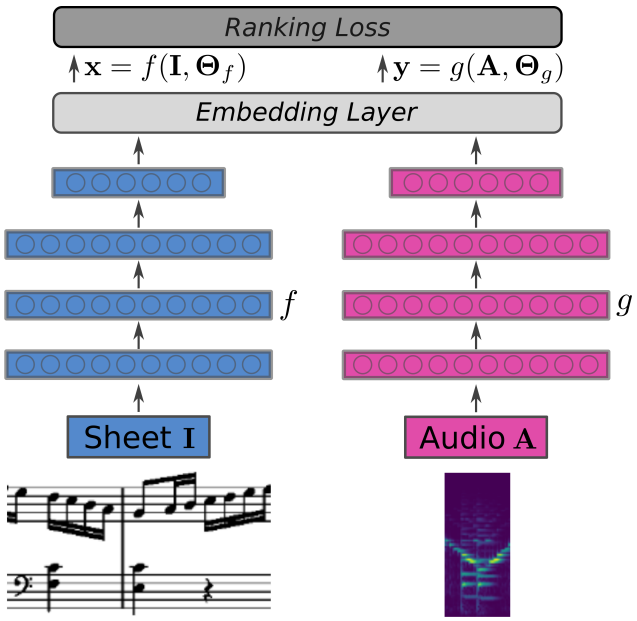


Figure 4.3: Architecture of correspondence learning network. The network is trained to optimize the similarity (in embedding space) between corresponding audio and sheet image snippets by minimizing a pair-wise ranking loss.

cessing the image part of an input pair and network g is responsible for processing the audio. The output of both networks (represented by the *Embedding Layer* in Figure 4.3) is a k-dimensional vector representation encoding the respective inputs. In our case the dimensionality of this representation is $k = 32$. We denote these hidden representations by $\mathbf{x} = f(\mathbf{I}, \mathbf{\Theta}_f)$ for the sheet image and $\mathbf{y} = g(\mathbf{A}, \mathbf{\Theta}_g)$ for the audio spectrogram, respectively, where $\mathbf{\Theta}_f$ and $\mathbf{\Theta}_g$ are the parameters of the two networks.

Given this network design, we now explain the pairwise ranking objective. Following Kiros et al. (2014) we first introduce a *scoring function* $s(\mathbf{x}, \mathbf{y})$ as the cosine similarity $\mathbf{x} \cdot \mathbf{y}$ between the two hidden representations ($\mathbf{x}$ and $\mathbf{y}$ are scaled to have unit norm). Based on this scoring function we optimize the following pairwise ranking objective ('hinge loss' (Rosasco et al., 2004)):

$$\mathcal{L}_{rank} = \sum_{\mathbf{x}} \sum_{k} \max\{0, \alpha - s(\mathbf{x}, \mathbf{y}) + s(\mathbf{x}, \mathbf{y}_k)\} \tag{4.1}$$

In our application $\mathbf{x}$ is an embedded sample of a sheet image snippet, $\mathbf{y}$ is the embedding of the matching audio excerpt and $\mathbf{y}_k$ are the embeddings of the *contrastive* (mismatching) audio excerpts (in practice all remaining samples of the

current training batch). When training our models we fix the mini-batch size to 100 samples. This means in particular, that for each $\mathbf{x}$ we are given one positive matching sample $\mathbf{y}$ and 99 contrastive samples $\mathbf{y}_k$. Mini-batches are drawn randomly from the entire training set without any sophisticated sampling strategy such as hard negative mining (Henriques et al., 2013). The hyper-parameter α defines the margin of the loss function and is set to 0.7 for all our experiments. The intuition behind this loss function is to encourage an embedding space where the distance between matching samples is lower than the distance between mismatching samples. If this condition is roughly satisfied, we can then perform cross-modality retrieval by simple nearest neighbour search in the embedding space. This will be explained in detail in Section 4.3.

The network itself is implemented as a VGG-style convolution network (Simonyan and Zisserman, 2015) consisting of 3×3 convolutions followed by 2×2 max-pooling as outlined in detail in Table 4.1. The final convolution layer computes 32 feature maps and is subsequently processed with a global average pooling layer (Lin et al., 2014) that produces a 32-dimensional vector for each input image and spectrogram, respectively. This is exactly the dimension of our retrieval embedding space. At the top of the network we put the canonically correlated embedding layer introduced in Chapter 6 (Dorfer et al., 2018d) and combine it with the ranking loss described above.

Before presenting a sheet-image snippet to the network, it is downsized by factor two ($160 \times 200 \rightarrow 80 \times 100$). This downsized image still contains all musically relevant content but reduces the number of computations required in the sheet image stack of the network. The saved computation time is then invested in doubling the number of feature maps to increase the capacity of our models. In terms of optimization we use the *Adam* update rule (Kingma and Ba, 2015) with an initial learning rate of 0.002. During training we watch the performance of the network on the validation set and halve the learning rate if there is no improvement for 30 epochs. This procedure is repeated five times to finetune the model.

4.3 Evaluation (1): Two-Way Snippet Retrieval

In this section, we evaluate the ability of our model to retrieve the correct counterpart when given an instance of the other modality as a search query. This first set of experiments is carried out on the lowest possible granularity, namely, on sheet image snippets and spectrogram excerpts such as shown in Figure 4.1.

For easier explanation we describe the retrieval procedure from an *audio query point of view* but stress that the opposite direction works in exactly the same fashion. Given a spectrogram excerpt $\mathbf{A}$ as a search query we want to retrieve the corresponding sheet image snippet $\mathbf{I}$. For retrieval preparation we first embed

Table 4.1: Audio - sheet music model. BN: Batch Normalization (Ioffe and Szegedy, 2015), ELU: Exponential Linear Unit (Clevert et al., 2016), MP: Max Pooling, Conv(3, pad-1)-16: 3×3 convolution, 16 feature maps and padding 1.

Sheet-Image 80 × 100	Audio (Spectrogram) 92 × 42
2×Conv(3, pad-1)-24	2× Conv(3, pad-1)-24
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-48	2× Conv(3, pad-1)-48
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-96	2× Conv(3, pad-1)-96
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-96	2× Conv(3, pad-1)-96
BN-ELU + MP(2)	BN-ELU + MP(2)
Conv(1, pad-0)-32-BN-LINEAR	Conv(1, pad-0)-32-BN-LINEAR
GlobalAveragePooling	GlobalAveragePooling
Embedding Layer + Ranking Loss	

all candidate image snippets $\mathbf{I}_j$ by computing $\mathbf{x}_j = f(\mathbf{I}_j)$ as the output of the image network. The candidate snippets originate from all unseen pieces from the respective test set. In a second step we embed the given query audio as $\mathbf{y} = g(\mathbf{A})$ using the audio pathway g of the network. Finally, we select the audio's nearest neighbour $\mathbf{x}^*$ from the set of embedded image snippets as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}_i} \left(1.0 - \frac{\mathbf{x}_i \cdot \mathbf{y}}{||\mathbf{x}_i||\,||\mathbf{y}||}\right) \tag{4.2}$$

based on their pairwise cosine distance. Figure 4.4 shows a sketch of this retrieval procedure.

4.3.1 Experimental Setup

We run retrieval experiments on all three training splits (compare Subsection 2.1.1 in Chapter 2) for the different combinations of data augmentation strategies described in Section 4.2.2. Results are presented for both retrieval direction, audio-to-sheet and sheet-to-audio retrieval. The unseen synthesizer and the tempo for the test set remain fixed for all settings. This allows us to directly investigate the influence of the different augmentation strategies. We further limit the number of retrieval test candidates to 2000 sheet snippets and audio excerpts respectively for all three splits. The 2000 candidates are randomly sampled across all of the test pieces. Having a fixed number of retrieval candidates makes performance of the learned models comparable across the different training splits.

As evaluation measures we compute the *Recall@k (R@k)*, the *Mean Reciprocal Rank (MRR)*, as well as the *Median Rank (MR)*. The *R@k* rate (high is better) is

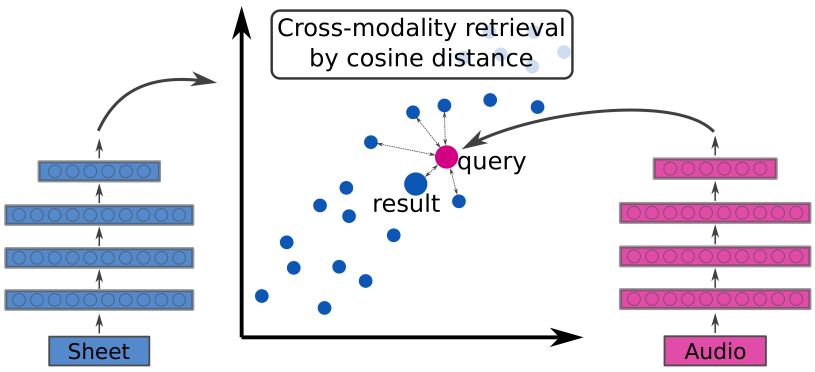


Figure 4.4: Sketch of sheet music-from-audio retrieval. The blue dots represent the embedded candidate sheet music snippets. The red dot is the embedding of an audio query. The larger blue dot highlights the closest sheet music snippet candidate selected as retrieval result.

the percentage of queries which have the correct corresponding counterpart in the first k retrieval results. The *MR* (low is better) is the median position of the target in a cosine-similarity-ordered list of available candidates. Finally, we define the MRR (higher is better) as the mean value of 1/rank over all queries where rank is again the position of the target in the similarity ordered list of available candidates.

4.3.2 Experimental Results

Table 4.2 summarizes the results on all three training splits for the different data augmentation strategies. Additionally, to get a better intuition of the results we provide the random-retrieval baseline for the 2000 candidates.

The common observation consistent across all datasets, performance measures and retrieval directions is that data augmentation helps to significantly improve the performance of all models. When isolating the effects of the two individual augmentation strategies, we see that audio augmentations yields the largest gain in performance on the test set. Surprisingly, sheet augmentation only helps to improve the performance on the bach-set and even degrades the model on the bach-out set. We do not report results on the validation set, but note that this behavior is reversed on the validation set. The reason for this is that the validation split is synthesized with a soundfont also covered by the training set. This means that in order to get a high performance on the validation set it is not required to generalize to unseen audio (spectrogram) characteristics. This is different for the test set, as it is synthesized with a hold out soundfont, explaining the large performance gain in Table 4.2 when applying audio augmentation. Finally, when combining both audio

bach-only

	Audio-to-Sheet Retrieval				Sheet-to-Audio Retrieval			
Aug.	**R@1**	**R@25**	**MRR**	**MR**	**R@1**	**R@25**	**MRR**	**MR**
none	0.25	0.73	0.37	6	0.34	0.81	0.46	3
sheet	0.38	0.81	0.49	3	0.45	0.85	0.57	2
audio	0.48	0.87	0.59	2	0.51	0.87	0.62	1
full	0.52	0.87	0.62	1	0.56	0.89	0.66	1
rand-bl	0.00	0.01	0.00	1000	0.00	0.01	0.00	1000

bach-out

	Audio-to-Sheet Retrieval				Sheet-to-Audio Retrieval			
Aug.	**R@1**	**R@25**	**MRR**	**MR**	**R@1**	**R@25**	**MRR**	**MR**
none	0.31	0.83	0.44	3	0.35	0.83	0.48	3
sheet	0.25	0.78	0.37	5	0.28	0.80	0.42	4
audio	0.38	0.83	0.50	2	0.39	0.85	0.52	2
full	0.46	0.86	0.57	2	0.46	0.87	0.57	2
rand-bl	0.00	0.01	0.00	1000	0.00	0.01	0.00	1000

all

	Audio-to-Sheet Retrieval				Sheet-to-Audio Retrieval			
Aug.	**R@1**	**R@25**	**MRR**	**MR**	**R@1**	**R@25**	**MRR**	**MR**
none	0.33	0.76	0.44	4	0.39	0.80	0.51	2
sheet	0.33	0.75	0.44	4	0.40	0.79	0.52	2
audio	0.46	0.82	0.57	2	0.49	0.84	0.59	2
full	0.50	0.83	0.60	2	0.51	0.85	0.61	1
rand-bl	0.00	0.01	0.00	1000	0.00	0.01	0.00	1000

Table 4.2: Snippet retrieval results. The table compares the influence of train/test splits and data augmentation on retrieval performance in both directions. For the audio augmentation experiments no sheet augmentation is applied and vice versa. *none* represents 1 sound font, with original tempo, and without sheet augmentation. We limit the number of retrieval candidates to 2000 for each of the splits to make the comparison across the different test sets fair.

and sheet augmentation we get the best results for all of the models generalizing to unseen scores as well as unseen audio. When recalling that our query length is only 42 spectrogram frames ($\approx$ 2 seconds of audio) per excerpt and that we select from a set of 2000 available candidate snippets, achieving a MR of not more than 2 is an

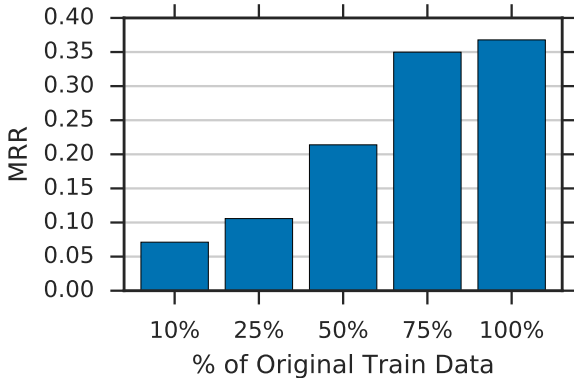


Figure 4.5: Influence of training set size on test set retrieval performance (MRR) evaluated on the bach-split in the no-augmentation setting.

impressive result. In particular, given a short excerpt of audio, the median position of the exactly matching counterpart is either 1 or 2 depending of the data split. This is even more impressive when keeping in mind that music is highly repetitive and that we consider only the exactly matching counterpart as a correct retrieval result. When comparing the two retrieval directions we see that sheet-to-audio retrieval works slightly but consistently better than the opposite directions again across all of the datasets.

In the following sections, we will see that this retrieval performance is sufficient for performing higher level tasks such as piece identification from audio queries. Furthermore, we will show in additional experiments in Section 4.5 that the resulting *full augmentation models* reach a level of generalization that makes them useful in practical real-world applications operating on scanned sheet images completely out of the synthetic training data domain.

4.3.3 Influence of Dataset Size

In this additional experiment we investigate the influence of training set size on the final retrieval performance. For this purpose we retrain the same network architecture once with 10, 25, 50 and 75 % of the original training examples in the no-augmentation setting of the bach-only split. We chose the no-augmentation setting for this experiment because we want to reveal the impact of the number of available training examples without cluttering the results with the effects of data augmentation.

Figure 4.5 compares the MRR on the test set for the respective proportions of training observations. The first, however not surprising observation, is that the training set size has a severe impact on the final retrieval capabilities of the model.

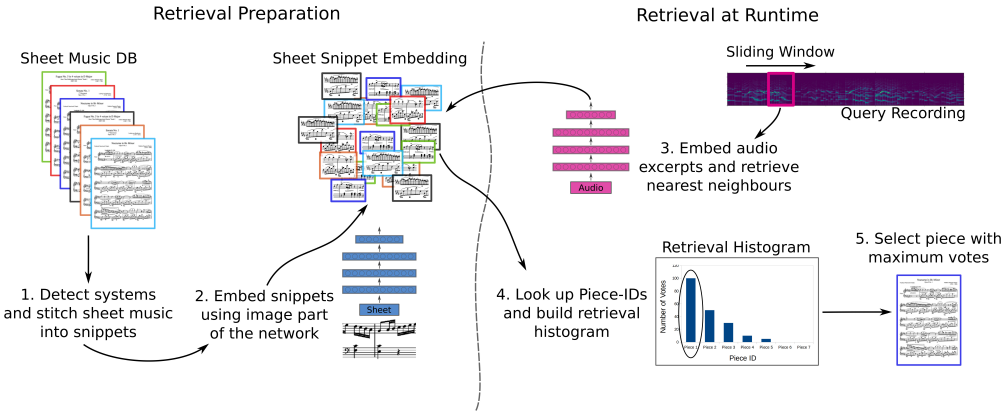


Figure 4.6: Piece retrieval concept from audio queries. The entire pipeline consists of two stages: retrieval preparation and retrieval at runtime (for details see Section 4.4).

The MRR increases for almost 30 points when comparing 10% to the data with the full dataset size. The second, more interesting observation, is that the relative improvement in performance is largest around 50% of the training set size (from 25% to 50% and from 50% to 75%). This indicates that there is a critical number of samples required to start generalizing to unseen sheet images. Finally, we observe that the gap between 75% and 100% of the data is fairly small compared to the remaining performance jumps. We interpret this as a positive outcome, suggesting that the full data set is sufficiently large enough to reach the full performance capabilities of the retrieval model.

4.4 Evaluation (2): Piece Identification and Performance Retrieval

Given the above model learning to express similarities between sheet music snippets and audio excerpts, we now describe how to use it for solving our targeted tasks: (1) identifying the respective piece of sheet music when given an entire audio recording as a query, and (2) given a score (sheet-image), retrieve a set of corresponding performances. The entire identification pipeline consists of two main stages summarized in Figure 4.6.

Score database preparation. Again, we describe the procedure from an audio query point of view and stress that the opposite direction works analogously. The first step is to prepare a sheet music retrieval database as follows: Given a set of sheet music images along with their annotated systems, we cut each piece of sheet

music j into a set of image snippets $\{\mathbf{I}_{ji}\}$ analogously to the snippets presented to our network for training. For each snippet, we store its originating piece j. We then embed all candidate image snippets into the retrieval embedding space by passing them through the image part f of the multimodal network. This yields, for each image snippet, a 32-dimensional embedding coordinate vector $\mathbf{x}_{ji} = f(\mathbf{I}_{ji})$. The left part of Figure 4.6 summarizes database preparation.

Retrieving sheet music at runtime. Once the database is prepared we perform piece retrieval as summarized in the right part of Figure 4.6. Given a whole audio recording as a search query, we aim to identify the corresponding piece of sheet music in our database.

First, we retrieve sheet snippets. As with the sheet image, we start by cutting the audio (spectrogram) into a set of excerpts $\{\mathbf{A}_1, ..., \mathbf{A}_K\}$, again exhibiting the same dimensions as the spectrograms used for training, and embed all query spectrogram excerpts $\mathbf{A}_k$ with the audio network g. Then we proceed as described in Section 4.3 and select for each audio its nearest neighbours from the set of all embedded image snippets. In our experiments we consider for each query excerpts its top 25 retrieval results for piece selection.

Second, we combine the retrieved snippets to select the pieces. Since we know for each of the image snippets its originating piece j, we can now have the retrieved image snippets $\mathbf{x}_{ji}$ *vote* for the piece. The piece achieving the highest count of votes is our final retrieval result. A similar procedure was used for example by Casey et al. (2008) for cover song identification.

4.4.1 Experimental Setup

We again carry out experiments on the three predefined data splits and compare the impact of data augmentation on the resulting retrieval (identification) performance. Results are presented for both retrieval direction, e.g., retrieving relevant performances given a score image as a search query and vice versa. It is also important to note that here we are still evaluating on our synthesized data. This will change in Section 4.5 where we work with scanned sheet music and recordings of real performances. As a retrieval measure, we compute the ranks@k ($Rk@k$) as the number of pieces retrieved within the first k retrieval results. Rk@1 means that a piece is ranked at position one and therefore identified correctly. To be consistent and comparable with Table 4.2 we also report the respective relative numbers ($R@k$) in brackets. Along with the data-splits we also report the number of candidate pieces (#) contained in the test set.

			Synthesized-to-Score			
Train Split	**#**	**Aug.**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
bach-only	50	none	33 (0.66)	46 (0.92)	48 (0.96)	2 (0.04)
		full	41 (0.82)	49 (0.98)	50 (1.00)	0 (0.00)
bach-out	173	none	125 (0.72)	158 (0.91)	163 (0.94)	10 (0.06)
		full	143 (0.83)	163 (0.94)	167 (0.97)	6 (0.03)
all	100	none	67 (0.67)	96 (0.96)	98 (0.98)	2 (0.02)
		full	82 (0.82)	97 (0.97)	99 (0.99)	1 (0.01)
			Score-to-Synthesized			
Train Split	**#**	**Aug.**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
bach-only	50	none	39 (0.78)	48 (0.96)	49 (0.98)	1 (0.02)
		full	47 (0.94)	50 (1.00)	50 (1.00)	0 (0.00)
bach-out	173	none	145 (0.84)	164 (0.95)	166 (0.96)	7 (0.04)
		full	149 (0.86)	169 (0.98)	172 (0.99)	1 (0.01)
all	100	none	94 (0.94)	98 (0.98)	99 (0.99)	1 (0.01)
		full	92 (0.92)	99 (0.99)	100 (1.00)	0 (0.00)

Table 4.3: Piece and performance identification results on synthetic data for all three data splits.

4.4.2 Experimental Results

Table 4.3 summarizes all piece identification results. The first observation is that the results regarding data augmentation are in line with the ones presented in Section 4.3. Looking at the different splits we see that a large fraction of the respective pieces is retrieved as the top retrieval result. When relaxing the retrieval measure and considering the Rk@5 we see that almost all of the pieces are contained in the set of top five results, especially in the direction of retrieving audio with a sheet image query. Although this is not the most sophisticated way of employing our network for piece retrieval, it clearly shows the usefulness of our model and its learned audio and sheet music representations for such tasks. Next steps towards making the identification process more robust will be to exploit the spatial and temporal structure (relation) of subsequent queries, such as proposed in (Balke et al., 2016).

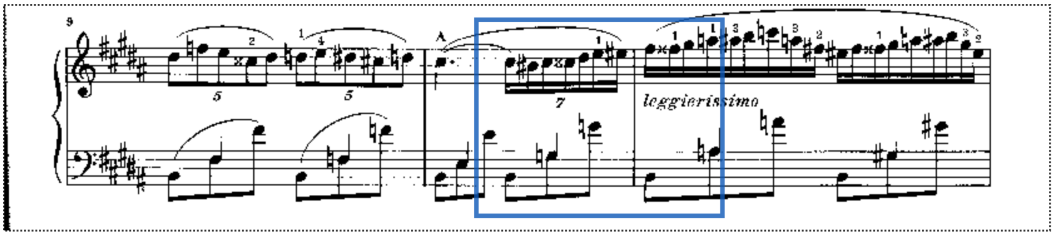


Figure 4.7: Exemplar staff line automatically extracted from a scanned score version of Chopin's Nocturne Op. 9 No. 3 in B major (Henle Urtext Edition; reproduced with permission). The blue box indicates an example sheet snippet fed to the image part of the retrieval embedding network.

4.5 Real-world Data: Retrieving Scanned Sheet Music and Real Performances

So far, both training the models and all of our experiments were carried out on synthetic data (rendered sheet music and synthesized MIDI performances). In this section, we present results on a set of additional experiments to answer the most prominent question: *How well do the models generalize to real data?*

4.5.1 Experimental Setup

Firstly, we clarify what we consider as real or realistic data in this context. Regarding sheet music, we use scanned images of scores from widely used commercial publishers such as Henle or Universal Edition. Figure 4.7 shows an example staff system from a piece by Frederic Chopin (Nocturne Op. 9 No. 3 in B major) to give an impression of this kind of data. For the performances, we use commercial audio recordings by various famous pianists (e.g., Ashkenazy, Pollini, Arrau, Horowitz, ...) that we happened to have in our music collection. We do not need any performance-to-score alignments if they are only used as test cases for piece retrieval. For further variability we have included music by different composers: Mozart (14 pieces; 88 score pages), Beethoven (29 pieces; 181 score pages), and Chopin (150 pieces; 871 score pages).

Retrieval preparation and retrieval itself follows exactly the descriptions outlined in Section 4.4 above. The sole difference in terms of data preparation is that for the scanned sheet music, we of course do not have the annotated system bounding boxes available. As the overall goal is to have the means to fully automatically index a large collection of scores, we developed an automatic system detection algorithm inspired by (Gallego and Calvo-Zaragoza, 2017; Dorfer et al., 2017b). Given the automatic system detection, we have all the tools to automatically create

the database (cf. Figure 4.6). Note that we do not need to detect noteheads – they were only relevant in aligning the modalities for training. For retrieval, we use the embedding networks trained on the *all* split using full data augmentation, as this data is most diverse in terms of sheet music and audio.

4.5.2 Experimental Results

Table 4.4 summarizes our results in the real data setting. To isolate the effects of real sheet images and real performance audios we repeat the experiment in two configurations: first with real scores and synthesized audios and second with real scores and real performances. Looking at the first group of experiments (top part of Table 4.4) with scanned sheet music and synthesized audios, we retrieve in the case of Mozart 13 of 14 as the top one candidate. The opposite retrieval direction works equally well. For Chopin we retrieve 127 out of 150 scanned scores at position one and 140 if we take the top five results into account. For the remaining sets and measures, we make similar observations. Given that the model was trained on purely rendered sheet music, with different and very consistent typesetting properties and containing no image noise at all, we consider this a remarkable result. We conclude that our model, in combination with the proposed dataset, is able to learn representations that generalize to completely unseen sheet music of a different typesetting style, beyond the synthetic training data.

In a final step, we further increase the level of difficulty of the retrieval setting. Instead of audios synthesized from MIDI, we use commercial recordings of performances by famous concert pianists. The bottom part of Table 4.4 lists our results in this configuration. Pieces and sheet music are identical to the experiments above to allow a direct comparison and to reveal the effects of the individual sources of potential problems. The general trend is that all performance measures drop compared to the synthetic audio settings. In terms of Rk@5 of Performance-to-Score retrieval, we are now able to retrieve 72 instead of 140 Chopin pieces, and 91 when considering Rk@10. Although this is a significant drop, it is in our opinion still a good result given the synthetic training data and the difficulty of the task. For the Mozart set, we are able to retrieve 12 out of 14 performances at position one given a scanned score as a query. Interestingly, the score-to-performance retrieval direction works better in this configuration for all three composers. We do not yet have a convincing explanation for this effect.

Based on these results we conclude that focusing on learning more robust audio representations is one of the main research challenges for future work (Section 4.7 and Part III contain a deeper discussion).

		Synthesized-to-Real-Score			
Composer	**#**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
Mozart	14	13 (0.93)	14 (1.00)	14 (1.00)	0 (0.00)
Beethoven	29	24 (0.83)	27 (0.93)	27 (0.93)	2 (0.07)
Chopin	150	127 (0.85)	140 (0.93)	145 (0.97)	5 (0.03)

		Real-Score-to-Synthesized			
Composer	**#**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
Mozart	14	13 (0.93)	14 (1.00)	14 (1.00)	0 (0.00)
Beethoven	29	25 (0.86)	27 (0.93)	29 (1.00)	0 (0.00)
Chopin	150	112 (0.75)	136 (0.91)	142 (0.95)	8 (0.05)

		Performance-to-Real-Score			
Composer	**#**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
Mozart	14	5 (0.36)	14 (1.00)	14 (1.00)	0 (0.00)
Beethoven	29	16 (0.55)	25 (0.86)	27 (0.93)	2 (0.07)
Chopin	150	36 (0.24)	72 (0.48)	91 (0.61)	59 (0.39)

		Real-Score-to-Performance			
Composer	**#**	**Rk@1**	**Rk@5**	**Rk@10**	**>Rk10**
Mozart	14	12 (0.86)	13 (0.93)	13 (0.93)	1 (0.07)
Beethoven	29	20 (0.69)	28 (0.97)	28 (0.97)	1 (0.03)
Chopin	150	58 (0.39)	94 (0.63)	111 (0.74)	39 (0.26)

Table 4.4: Evaluation on real data: Piece retrieval results on scanned sheet music and recordings of real performances. The model used for retrieval is trained on the all-split with full data augmentation.

4.6 Audio-to-Sheet-Music Alignment

As a second usage scenario for our approach we present the task of audio-to-sheet-music alignment. Here, the goal is to align a performance (given as an audio file) to its respective score (as images of the sheet music), i.e., computing the corresponding location in the sheet music for each time point in the performance, and vice versa.

For computing the actual alignments we rely on Dynamic Time Warping (DTW), which is a standard method for sequence alignment (Rabiner and Juang, 1993), and is routinely used in the context of music processing (Müller, 2015). Generally, DTW takes two sequences as input and computes an optimal non-linear alignment between them, with the help of a local cost measure that relates points of the two sequences to each other.

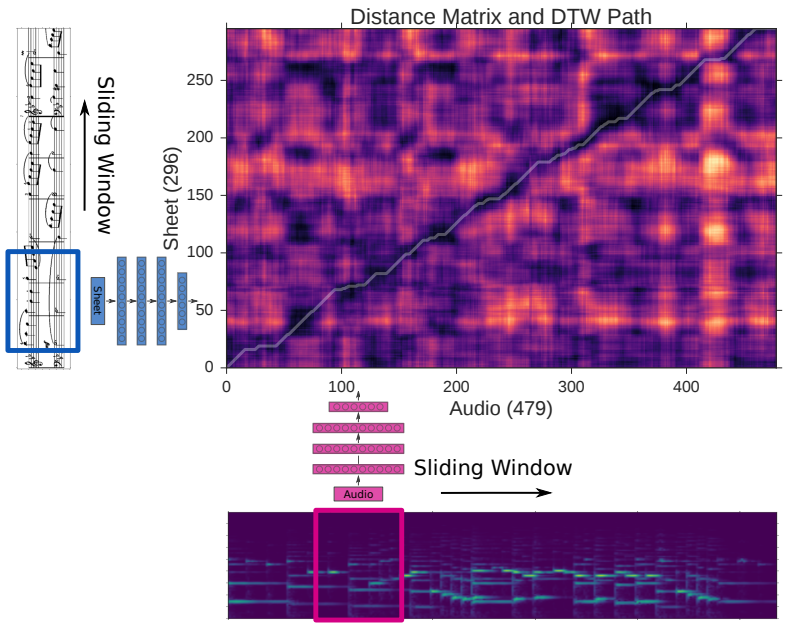


Figure 4.8: Sketch of audio-to-sheet-music alignment by DTW on a similarity matrix computed on the embedding representation learned by the multimodal network. The white line highlights the path of minimum costs through the sheet music given the audio.

For our task the two sequences to be aligned are the sequence of snippets from the sheet music image and the sequence of audio (spectrogram) excerpts, as described in Section 4.2.1. The neural network presented in Section 4.2.3 is then used to derive a local cost measure by computing the pairwise cosine distances between the embedded sheet snippets and audio excerpts (see Equation 4.2). The resulting cost matrix that relates all points of both sequences to each other is shown in Figure 4.8, for a short excerpt from a simple Bach minuet. We also indicate the sliding window retrieval embedding space alignment procedure with the respective part of the multimodal network. Then, the standard DTW algorithm is used to obtain the optimal alignment path.

4.6.1 Experimental Setup

We again evaluate the alignment procedure on all of the tree splits using the models produced in the full augmentation setup. As evaluation measure we compute the absolute *alignment error* (distance in pixels) of the estimated alignment to its ground truth alignment for each of the sliding window positions. We further nor-

malize the errors by dividing them by the sheet image width to be independent of image resolution. As a naive baseline we compute a linear interpolation alignment which would correspond to a straight line diagonal in the distance matrix in Figure 4.8. We consider this as a valid reference as we do not consider repetitions for our experiments, yet (in which case things would become somewhat more complicated). We further emphasize that the purpose of this experiment is to provide a proof of concept for this class of models in the context of sheet music alignment tasks, not to compete with existing specialized algorithms for music alignment.

4.6.2 Experimental Results

All results of this experiment are summarized by the boxplots in Figure 4.9 comparing the linear baseline with DTW in embedding space. The median alignment error for the linear baseline is 9.46% of the image widths for the bach-split, 12.10% for the bach-out split and 8.62% for the all-split. When computing a DTW path through the distance matrix inferred by our multimodal audio-sheet-music network this error decreases to 1.2% on the bach-split which is approximately 2.5mm in a printed page of sheet music (3.5mm for bach-out and 3.0mm for all-split). When looking at the box plots we see that the proposed alignment procedure produces consistent results across all of the splits even though the linear baseline shows a different behavioral on the respective splits. The most important observation however is the variance of the alignment errors. For the embedding-DTW the alignment error stays below 5% of the image width for almost all of the samples ($\approx$ 10mm in a printed sheet). These results indicate a very robust alignment behavior.

4.7 Discussion and Future Work

Our experiments on piece and performance identification on both synthetic and real data (see Sections 4.4 and 4.5) lead to the following observations: Given the MSMD data set and the proposed methodology, we can learn retrieval models that clearly generalize beyond the synthetic training data domain. This holds especially for scanned images of unseen sheet music. When dealing with real performances, we still achieve good retrieval results, but encounter a significant drop in all performance measures. We have to remember that our model (a multimodal convolutional neural network) has a fixed and limited field of view on both the audio (excerpt) and the sheet music (snippet). While this is not a problem on the sheet music side, it definitely is for performance audios, which may exhibit rather extreme tempo changes and differences (in addition to challenges such as asynchronous onsets, pedal, room acoustics, or dynamics). Given these facts about performance and our experimental findings, we believe that learning robust audio representations is one of the main open research problems to be addressed.

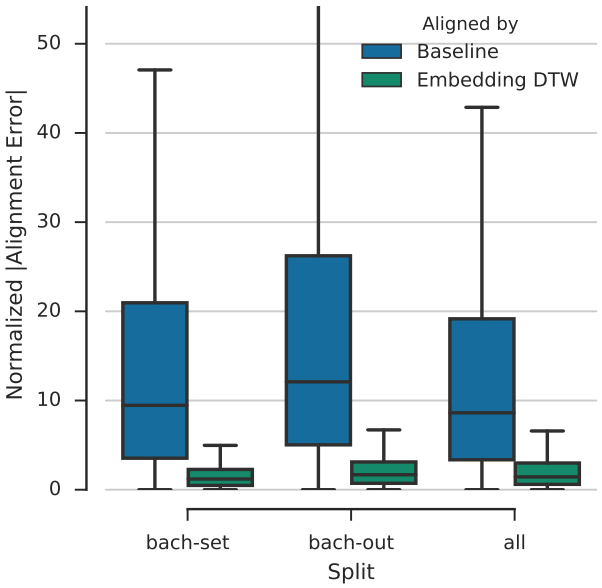


Figure 4.9: Absolute alignment errors normalized by sheet image width. We compare the linear baseline with a DTW on the cross-modal distance matrix computed on the embedded sheet snippets and spectrogram excerpts.

Regarding the retrieval (piece/performance identification) methodology, note that so far we completely ignore the strong temporal dependencies between subsequent queries, which are inherent in music. An obvious next step will be to extend the identification procedure in a way that exploits these spatio-temporal relationships (e.g., as in (Balke et al., 2016)).

4.8 Conclusion

This chapter presented a methodology for learning correspondences between short snippets of sheet music and their counterparts in the music audio. The learned shared latent representation of the two modalities can be utilized for cross-modality retrieval, i.e., for identifying scores from full audio queries and vice versa. As a second application scenario, we showed how to utilize the joint embedding space for an offline alignment of audio and sheet music using DTW. We make all our experimental code (including pre-trained embedding models) freely available, hoping to reduce the initial hurdles for working with this kind of data. Finally, we showed that the proposed methodology in combination with the MSMD dataset leads to models that are beginning to generalize to real-world retrieval scenarios with scanned sheet music and real performance audios.

5 Learning to Listen, Read, and Follow: Score Following as a Reinforcement Learning Game

In the final chapter of this part we will employ the third machine learning paradigm visited in this thesis, deep reinforcement learning. We will also revisit the task of score following in sheet music images (see Chapter 3), the process of tracking a musical performance (audio) with respect to a known symbolic representation (a score). We start this chapter by formulating score following as a multimodal Markov Decision Process (MDP), the mathematical foundation for sequential decision making. Given this formal definition, we address the score following task with state-of-the-art deep reinforcement learning (RL) algorithms such as synchronous advantage actor critic (A2C). In particular, we design multimodal RL agents that simultaneously learn to listen to music, read the scores from images of sheet music, and follow the audio along in the sheet, in an end-to-end fashion. All this behavior is learned entirely from scratch, based on a weak and potentially delayed reward signal that indicates to the agent how close it is to the correct position in the score. Besides discussing the theoretical advantages of this learning paradigm, we show in experiments that it is in fact superior compared to previously discussed methods (Chapter 3, Dorfer et al. (2016b)) for score following in raw sheet music images. The experimental evaluations presented in this chapter are again carried out on the MSMD dataset introduced in Chapter 2.

This chapter is based on the following publication:

- M. Dorfer, F. Henkel, and G. Widmer. Learning to listen, read, and follow: Score following as a reinforcement learning game (**Best Paper and Best Poster Award**). In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018c.

Personal Contributions I had the idea of formulating score following as a reinforcement learning game, implemented the prototype and did the experimental evaluations. Florian was a great help with implementing the reinforcement learning framework and the final version of the game.

Erratum During the final review of my thesis I recognized a mistake in the evaluation of the baseline method MM-Loc which effects the results reported in Table 5.2. I corrected the error and updated the table accordingly (marked as MM-Loc*) but also keep the original results published in the corresponding paper. Note that this does not effect the contributions of this chapter and the work presented in (Dorfer et al., 2018c). On the contrary, the original evaluation of MM-Loc was too optimistic and after updating it the advantages of our method, in terms of tracking stability, are even more emphasized.

5.1 Introduction

This chapter returns to the a task already addressed in Chapter 3: the problem of score following in sheet music images. The task of an automatic score following system is to follow a musical performance with respect to a known symbolical representation, the score. Figure 5.1 shows a sketch of this process. In contrast to

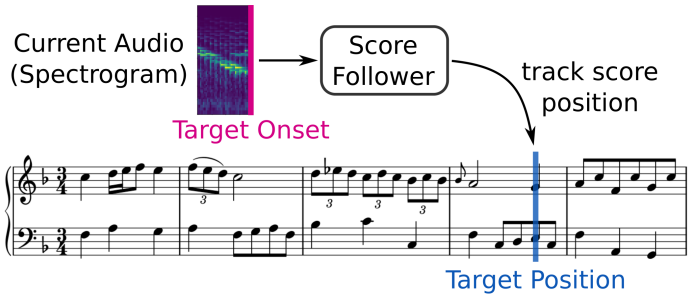


Figure 5.1: Sketch of score following in sheet music. Given the incoming audio, the score follower has to track the corresponding position in the score (image).

audio-score alignment in general (cf. Section 4.6, Müller (2015)), all of this takes place in an on-line fashion. Score following itself has a long history in Music Information Retrieval (MIR) and forms the basis for many subsequent applications such as automatic page turning (Arzt et al., 2008), automatic accompaniment (Cont, 2009; Raphael, 2010) or the synchronization of visualizations to the live music during concerts (Arzt et al., 2015; Prockup et al., 2013).

Traditional approaches to the task depend on a symbolic, computer-readable representation of the score, such as MusicXML or MIDI (see e.g. (Kurth et al., 2007; Nakamura et al., 2015; Arzt et al., 2015; Prockup et al., 2013; Cont, 2009; Raphael, 2010; Miron et al., 2014; Duan and Pardo, 2011; Izmirli and Sharma, 2012)). This representation is created either manually (e.g. via the time-consuming process of (re-)setting the score in a music notation program), or automatically via

optical music recognition software (Hajič jr and Pecina, 2017; Byrd and Simonsen, 2015; Balke et al., 2015). However, automatic methods are still unreliable and thus of limited use, especially for more complex music like orchestra pieces (Thomas et al., 2012).

To avoid these complications, we introduced a multimodal deep neural network that directly learns to match sheet music and audio in an end-to-end fashion (Chapter 3, (Dorfer et al., 2016b)). Given short excerpts of audio and the corresponding sheet music, the network learns to predict which location in the given sheet image best matches the current audio excerpt. In this setup, score following can be formulated as a multimodal localization task. However, one problem with this approach is that successive time steps are treated independently from each other. We will see in our experiments that this causes jumps in the tracking process especially in the presence of repetitive passages. Also related to this is the approach presented in Chapter 4 where we train a multimodal neural network to learn a joint embedding space for snippets of sheet music and corresponding short excerpts of audio. The learned embedding allows to compare observations across modalities, e.g., via their cosine distance. This learned cross-modal similarity measure is then used to compute an off-line alignment between audio and sheet music via dynamic time warping (see Section 4.6).

The proposal presented in this chapter is inspired by these approaches, but uses a fundamentally different machine learning paradigm. The central idea is to interpret score following as a *multimodal control problem* (Duan et al., 2016) where the agent has to navigate through the score by adopting its reading speed in reaction to the currently playing performance. To operationalize this notion, we formulate score following as a *Markov Decision Process (MDP)* in Section 5.3. MDPs are the mathematical foundation for sequential decision making and permit us to address the problem with state-of-the-art *Deep Reinforcement Learning (RL)* algorithms (Section 5.4). Based on the MDP formulation, we design agents that consider both the score and the currently playing music to achieve an overall goal, that is to track the correct position in the score for as long as possible. This kind of interaction is very similar to controlling an agent in a video game, which is why we term our MDP the *score following game*; it is in fact inspired by the seminal paper by Mnih et al. (Mnih et al., 2015) which made a major contribution to the revival of deep RL by achieving impressive results in a large variety of Atari games. In experiments with monophonic as well as polyphonic music (Section 5.5), we will show that the RL approach is indeed competitive with previously proposed score following methods (Chapter 3, Dorfer et al. (2016b)). The code for both the score following game as well as the corresponding multimodal RL agents is available at `https://github.com/CPJKU/score_following_game`.

5.2 Description of Data

To set the stage, we first need to describe the kind of data needed for training and evaluating the multimodal RL score following agents. As the data representation are in line with the previous chapters we limit ourself to briefly recapitulating the relevant parts. Building on top of the MSMD dataset introduced in Chapter 2, we are given a collection of piano pieces represented as pairs of audio recordings and sheet music images. In order to train our models and to later quantify the score following error, we again require correspondences between individual pixel locations of the note heads in a sheet and their respective counterparts (note onset events) in the respective audio recordings. This is exactly the kind of data and annotations included in the MSMD dataset so everything is ready to develop our agents. For the experiments in this chapter we will rely on the all-split as introduced in Section 2.1.1. For a detailed description of the entire alignment process we refer to Chapter 2.

5.3 Score Following as a Markov Decision Process

Reinforcement learning can be seen as a computational approach to learning from interaction to achieve a certain predefined goal (Sutton and Barto, 1998). In this section, we formulate the task of score following as a *Markov Decision Process (MDP)*, the mathematical foundation for reinforcement learning or, more generally, for the problem of sequential decision making[1]. Figure 5.2 provides an overview of the components involved in the score following MDP.

The *score following agent* (or *learner*) is the active component that interacts with its *environment*, which in our case is the score following game. The interaction takes place in a closed loop where the environment confronts the agent with a new situation (a *state* S_t) and the agent has to respond by making a decision, selecting one out of a predefined set of possible *actions* A_t. After each action taken the agent receives the next state S_{t+1} and a numerical *reward signal* R_{t+1} indicating how well it is doing in achieving the overall goal. Informally, the agent's goal in our case is to track a performance in the score as accurately and robustly as possible; this criterion will be formalized in terms of an appropriate reward signal in Section 5.3.3 below. By running the MDP interaction loop we end up with a sequence of states, actions and rewards $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ...$, which is the kind of experience a RL agent is learning its behavior from. We will elaborate on different variants of the learning process in Section 5.4. The remainder of this section specifies all components of the score following MDP in detail. In practice,

[1] The notation and general review of reinforcement learning in this chapter follows the book by Sutton and Barto (1998).

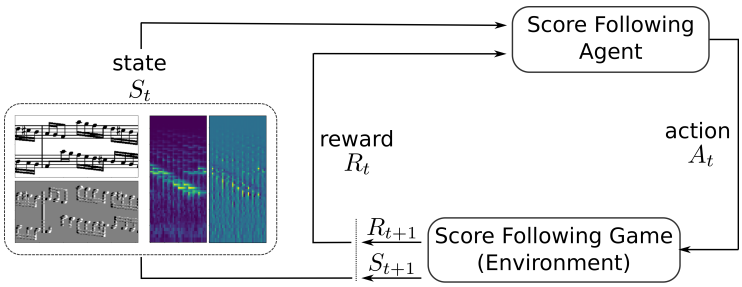


Figure 5.2: Sketch of the score following MDP. The agent receives the current state of the environment S_t and a scalar reward signal R_t for the action taken in the previous time step. Based on the current state it has to choose an action (e.g. decide whether to increase, keep or decrease its speed in the score) in order to maximize future reward by correctly following the performance in the score.

our MDP is implemented as an environment in OpenAI-Gym[2], an open source toolkit for developing and comparing reinforcement learning algorithms.

5.3.1 Score Following Markov States

Our agents need to operate on two different inputs at the same time, which together form the state S_t of the MDP: input modality one is a sliding window of the sheet image of the current piece, and modality two is an audio spectrogram excerpt of the most recently played music (~ 2 seconds). Figure 5.3 shows an example of this input data for a piece by J.S. Bach. Given the audio excerpt as an input the agent's task is to navigate through the global score to constantly receive sheet windows from the environment that match the currently playing music. How this interaction with the score takes place is explained in the next subsection. The important part for now is to note that score following embodies dynamics which have to be captured by our state formulation, in order for the process to satisfy the Markov property. Therefore, we extend the state representation by adding the one step differences (Δ) of both the score and the spectrogram. With the Δ images and spectrograms a state contains all the information needed by the agent to determine where and how fast it is moving along in the sheet image.

5.3.2 Agents, Actions and Policies

The next item in the MDP (Figure 5.2) is the agent, which is the component interacting with the environment by taking actions as a response to states received.

[2] https://gym.openai.com/

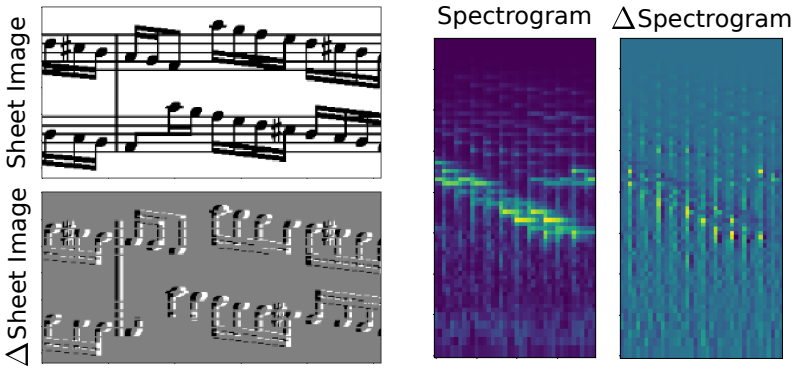


Figure 5.3: Markov state of the score following MDP: the current sheet sliding window and spectrogram excerpt. To capture the dynamics of the environment we also add the one step differences (Δ) wrt. the previous time step (state).

As already mentioned, we interpret score following as a multimodal control problem where the agent decides how fast it would like to progress in the score. In more precise terms, the agent controls its score progression speed v_{pxl} in *pixels per time step* by selecting from a set of actions $A_t \in \{-\Delta v_{pxl}, 0, +\Delta v_{pxl}\}$ after receiving state S_t in each time step. Actions $\pm\Delta v_{pxl}$ increase or decrease the speed by a value of Δv_{pxl} pixels per time step. Action $a_1 = 0$ keeps it unchanged. To give an example: a pixel speed of $v_{pxl} = 14$ would shift the sliding sheet window 14 pixels forward (to the right) in the global unrolled score.

Finally, we introduce the concept of a *policy* $\pi_\Theta(a|s)$ to define an agent's behavior. π is a conditional probability distribution over actions conditioned on the current state. Given a state s, it computes an action selection probability $\pi_\Theta(a|s)$ for each of the candidate actions $a \in A_t$. The probabilities are then used for sampling one of the possible actions. In Section 5.4 we will show how to use deep neural networks as function approximators for policy π_Θ by optimizing the parameters Θ of a policy network.

5.3.3 Goal Definition: Reward Signal and State Values

In order to learn a useful action selection policy, the agent needs *feedback*. This means that we need to define how to report back to the agent how well it does in accomplishing the task and, more importantly, what the task actually is.

The one component in an MDP that defines the overall goal is the reward signal $R_t \in \mathbb{R}$. It is provided by the environment in form of a scalar, each time the agent performs an action. The *sole objective of a RL agent is to maximize the cumulative reward over time*. Note, that achieving this objective requires fore-

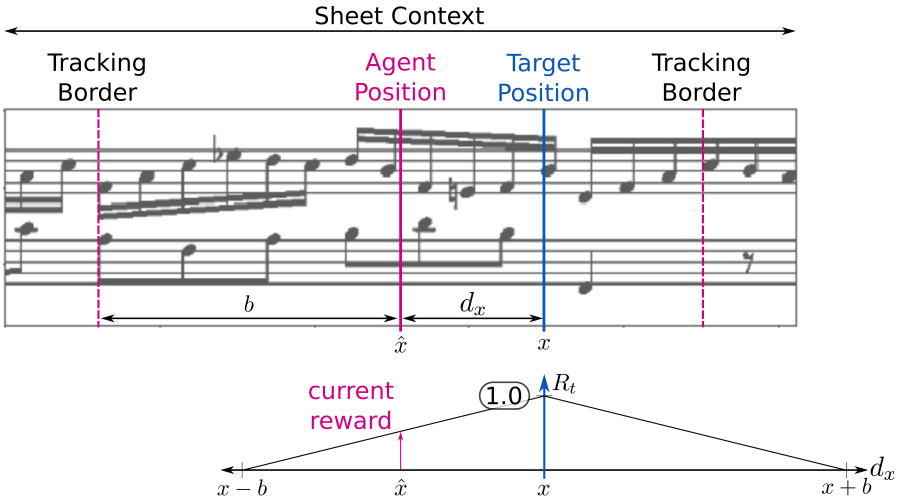


Figure 5.4: Reward definition in the score following MDP. The reward R_t decays linearly (range [0, 1]) depending on the agent's distance d_x to the current true score position x.

sight and planning, as actions leading to high instantaneous reward might lead to unfavorable situations in the future. To quantify this longterm success, RL introduces the *return* G which is defined as the discounted cumulative future reward: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$. The discount rate γ (with $0.0 < \gamma \leq 1.0$, in our case 0.9) is a hyper-parameter assigning less weight to future rewards if smaller than 1.0.

Figure 5.4 summarizes the reward computation in our score following MDP. Given annotated training data as described in Section 5.2, the environment knows, for each onset time in the audio, the true target position x in the score. From this, and the current position $\hat{x}$ of the agent, we compute the current tracking error as $d_x = \hat{x} - x$, and define the reward signal r within a predefined *tracking window* $[x - b, x + b]$ around target position x as: $r = 1.0 - |d_x|/b$. Thus, the reward per time step reaches its maximum of 1.0 when the agent's position is identical to the target position, and decays linearly towards 0.0 as the tracking error reaches the maximum permitted value b given by the window size. Whenever the absolute tracking error exceeds b (the agent drops out of the window), we reset the score following game (back to start of score, first audio frame). As an RL agent's sole objective is to maximize cumulative future reward, it will learn to match the correct position in the score and to not lose its target by dropping out of the window. We define the target onset, corresponding to the target position in the score, as the *rightmost frame* in the spectrogram excerpt. This allows to run the agents on-line, introducing only the delay required to compute the most recent spectrogram frame. In practice,

we linearly interpolate the score positions for spectrogram frames between two subsequent onsets in order to produce a continuous and stronger learning signal for training.

As with policy π, we will use function approximation to predict the future cumulative reward for a given state s, estimating how good the current state actually is. This estimated future reward is termed the *value* $V(s)$ of state s. We will see in the next section how state-of-the-art RL algorithms use these value estimates to stabilize the variance-prone process of policy learning.

5.4 Learning to Follow

Given the formal definition of score following as an MDP we now describe how to address it with reinforcement learning. Note that there is a large variety of RL algorithms. We focus on *policy gradient methods*, in particular the class of *actor-critic methods*, due to their reported success in solving control problems (Duan et al., 2016). The learners utilized are *REINFORCE with Baseline* (Williams, 1992) and *Synchronous Advantage Actor Critic (A2C)* (Mnih et al., 2016; Wu et al., 2017), where the latter is considered a state-of-the-art approach. As describing the methods in full detail is beyond the scope of this chapter and thesis, we provide an intuition on how the methods work and refer the reader to the respective papers.

5.4.1 Policy and State-Value Approximation via DNNs

In Section 5.3, we introduced *policy* π_Θ, determining the behavior of an agent, and *value function* $V(s)$, predicting how good a certain state s is with respect to cumulative future reward. Actor-critic methods make use of both concepts. The actor is represented by policy π_Θ and is responsible for selecting the appropriate action in each state. The critic is represented by the value function $V(s)$ and helps the agent to judge how good the selected actions actually are. In the context of deep RL both functions are approximated via a neural network, termed policy and value network. We denote the parameters of the policy network with Θ in the following.

Figure 5.5 shows a sketch of such a network architecture. As in Chapter 3, we use a multimodal convolutional neural network operating on both sheet music and audio at the same time. The input to the network is exactly the Markov state of the MDP introduced in Section 5.3.1. The left part of the net processes sheet images, the right part spectrogram excerpts (including the Δ images). After low-level representation learning, the two modalities are merged by concatenation and further processed using dense layers. This architecture implies that policy and value network share the parameters of the lower layers, which is a common choice in RL (Mnih et al., 2016). Finally, there are two output layers: the first represents our policy and

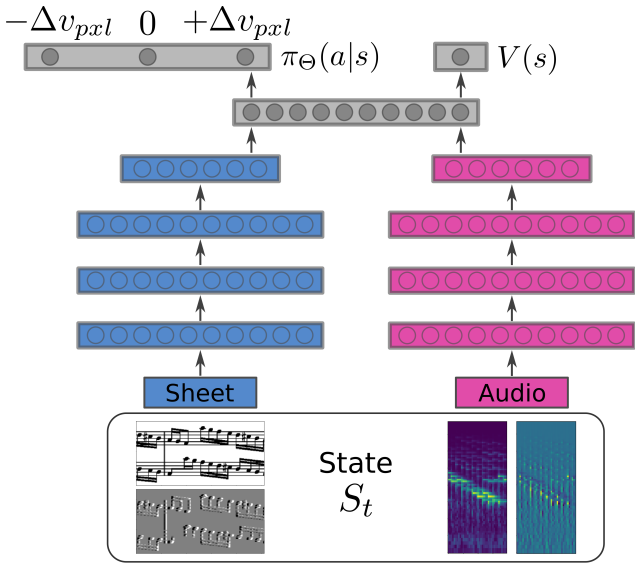


Figure 5.5: Multimodal network architecture used in the score following agents. Given state s the policy network predicts the action selection probability $\pi_\Theta(a|s)$ for the allowed action $A_t \in \{-\Delta v_{pxl}, 0, +\Delta v_{pxl}\}$. The value network, sharing parameters with the policy network, provides a state-value estimate $V(s)$ for the current state. The lower network layers are shared between π_Θ and V.

predicts the action selection probability $\pi_\Theta(a|s)$. It contains three output neurons (one for each possible action) converted into a valid probability distribution via soft-max activation. The second output layer consists of one linear output neuron predicting the value $V(s)$ of the current state. Table 5.1 lists the exact architectures used for our experiments. We use exponential linear units (Clevert et al., 2016) for all but the two output layers.

5.4.2 Learning a Policy via Actor-Critic

One of the first algorithms proposed for optimizing a policy was REINFORCE (Williams, 1992), a Monte-Carlo algorithm that learns by generating entire episodes $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, ...$ of states, actions and rewards by following policy π_Θ while interacting with the environment. Given this sequence it updates the parameters Θ of the policy network according to the following update rule by replaying the episode time step by time step:

$$\Theta \leftarrow \Theta + \alpha G_t \nabla_\Theta \ln \pi_\Theta(A_t|S_t, \Theta) \tag{5.1}$$

Table 5.1: Network architecture. DO: Dropout, Conv(3, stride-1)-16: 3×3 convolution, 16 feature maps and stride 1.

Audio (Spectrogram) 78×40	Sheet-Image 80×256
Conv(3, stride-1)-32	Conv(5, stride-(1, 2))-32
Conv(3, stride-1)-32	Conv(3, stride-1)-32
Conv(3, stride-2)-64	Conv(3, stride-2)-64
Conv(3, stride-1)-64 + DO(0.2)	Conv(3, stride-1)-64 + DO(0.2)
Conv(3, stride-2)-64	Conv(3, stride-2)-64
Conv(3, stride-2)-96	Conv(3, stride-2)-64 + DO(0.2)
Conv(3, stride-1)-96	Conv(3, stride-2)-96
Conv(1, stride-1)-96 + DO(0.2)	Conv(1, stride-1)-96 + DO(0.2)
Dense(512)	Dense(512)
Concatenation + Dense(512)	
Dense(256) + DO(0.2)	Dense(512) + DO(0.2)
Dense(3) - Softmax	Dense(1) - Linear

α is the step size or learning rate and G_t is the true discounted cumulative future reward (the return) received from time step t onwards. Gradient ∇_Θ is the direction in parameter space in which to go if we want to maximize the selection probability of the respective action. This means whenever the agent did well, achieving a high return G_t, we take larger steps in parameter space towards selecting the responsible actions. By changing the parameters of the policy network, we of course also change our policy (behavior) and we will select beneficial actions more frequently in the future when confronted with similar states.

REINFORCE and policy optimization are known to have high variance in the gradient estimate (Greensmith et al., 2004). This results in slow learning and poor convergence properties. To address this problem, **REINFORCE with Baseline** (REINFORCE$_{bl}$) adapts the update rule of Equation (5.1) by subtracting the estimated state value $V(s)$ (see Section 5.3.3) from the actual return G_t received:

$$\Theta \leftarrow \Theta + \alpha(G_t - V(s))\nabla_\Theta \ln \pi_\Theta(A_t|S_t, \Theta) \tag{5.2}$$

This simple adaptation helps to reduce variance and improve convergence. The value network itself is learned by minimizing the mean squared error between the actually received return and the predicted value estimate of the network, $(G_t - V(s))^2$. REINFORCE$_{bl}$ will be the first learning algorithm considered in our experiments.

Actor-critic methods are an extension of the baseline concept, allowing agents to learn in an online fashion while interacting with the environment. This avoids the need for creating entire episodes prior to learning. In particular, our actor-critic agent will only look into the future a fixed number of t_{max} time steps (in our case, 15). This implies that we do not have the actual return G_t available for updating the value function. The solution is to *bootstrap* the value function (i.e., update

the value estimate with estimated values), which is the core characteristic of actor-critic methods. Mnih et al. (2016) propose the **Synchronous Advantage Actor Critic (A2C)** and show that running multiple actors (in our case 16) in parallel on different instances of the same kind of environment, further helps to stabilize training. We will see in our experiments that this also holds for the score following task. For a detailed description of the learning process we refer to the original paper.

5.5 Experimental Results

In this section we experimentally evaluate our RL approach to score following and compare it to the method introduced in Chapter 3 addressing the same task. In addition to quantitative analysis we also provide a video of our agents interacting with the score following environment[3].

5.5.1 Experimental Setup

Two different datasets will be used in our experiments. The *Nottingham Dataset* containing monophonic melodies of folk music (training: 187, validation: 63, testing: 46); it was already used in Chapter 3 to evaluate our first proposal to score following in sheet music images. The second dataset is the all-split of the MSMD dataset introduced in Chapter 2 (training: 360, validation: 19, testing: 100). It covers polyphonic music and is a substantially harder challenge to a score follower. In both cases the sheet music is typeset with Lilypond and the audios are synthesized from MIDI using an acoustic piano sound font. This automatic rendering process provides the precise audio – sheet music alignments required for training (see Section 5.2). For audio processing we set the computation rate to 20 FPS and compute log-frequency spectrograms at a sample rate of 22.05kHz. The FFT is computed with a window size of 2048 samples and post-processed with a logarithmic filterbank allowing only frequencies from 60Hz to 6kHz (78 frequency bins).

The spectrogram context visible to the agents is set to 40 frames (2 sec. of audio) and the sliding window sheet images cover 160×512 pixels and are further downscaled by a factor of two before being presented to the network. As optimizer we use the *Adam* update rule (Kingma and Ba, 2015) with an initial learning rate of 10^{-4} and running average coefficients of 0.5 and 0.999. We then train the models until there is no improvement in the number of tracked onsets on the validation set for 50 epochs and reduce the learning rate by factor 10 three times. The tempo change action Δv_{pxl} is 0.5 for Nottingham and 1.0 pixel per time step for the polyphonic pieces of MSMD.

[3] score following video: https://youtu.be/COPNciY510g

5.5.2 Evaluation Measures and Baselines

Recall from Section 5.3.3 and Figure 5.4 that from the agent's position $\hat{x}$ and the ground truth position x, we compute the tracking error d_x. This error is the basis for our evaluation measures. However, compared to training, we only consider time steps in our evaluation where there is actually an onset present in the audio. While interpolating intermediate time steps is helpful for creating a stronger learning signal (Section 5.3.3), it is not musically meaningful. Specifically, we will report the evaluation statistics *mean absolute tracking error* $\overline{|d_x|}$ as well as its standard deviation $std(|d_x|)$ over all test pieces. These two measures quantify the *accuracy* of the score followers. To also measure their *robustness* we compute the ratio R_{on} of overall tracked onsets as well as the ratio of pieces R_{tue} tracked from beginning entirely to the end.

As *baseline method* we consider the approach described in Chapter 3 (Dorfer et al., 2016b), which models score following as a multimodal localization task (denoted by MM-Loc in the following).

As a *second baseline*, we also tried to train an agent to solve the score following MDP in a fully supervised fashion. This is theoretically possible, as we know for each time point the exact corresponding position in the score image, which permits us to derive an optimal tempo curve and, consequently, an optimal sequence of tempo changes for each of the training pieces. Figure 5.6 shows such an optimal tempo curve along with the respective tempo change actions for a short Bach piece. The latter would serve as targets y in a supervised regression problem $y = f(x)$. The network structure we used for this experiment is identical to the one in Figure 5.5 except for the output layers. Instead of policy π_θ and value V we only keep a single linear output neuron predicting the value of the optimal tempo change in each time step. However, a closer look at Figure 5.6 already reveals the problem inherent in this approach. The optimal tempo change is close to zero most of the time. For the remaining time steps we observe sparse spikes of varying amplitude. When trying to learn to approximate these optimal tempo changes (with a mean squared error optimization target), we ended up with a network that predicts values very close to zero for all its inputs. We conclude that the relevant tempo change events are too sparse for supervised learning and exclude the method from our tables in the following. Besides these technical difficulties we will also discuss conceptual advantages of addressing score following as an MDP in Section 5.6.

5.5.3 Experimental Results

Table 5.2 provides a summary of the experimental results. Looking at the Nottingham dataset, we observe large gaps in performance between the different approaches. Both RL based methods manage to follow almost all of the test pieces

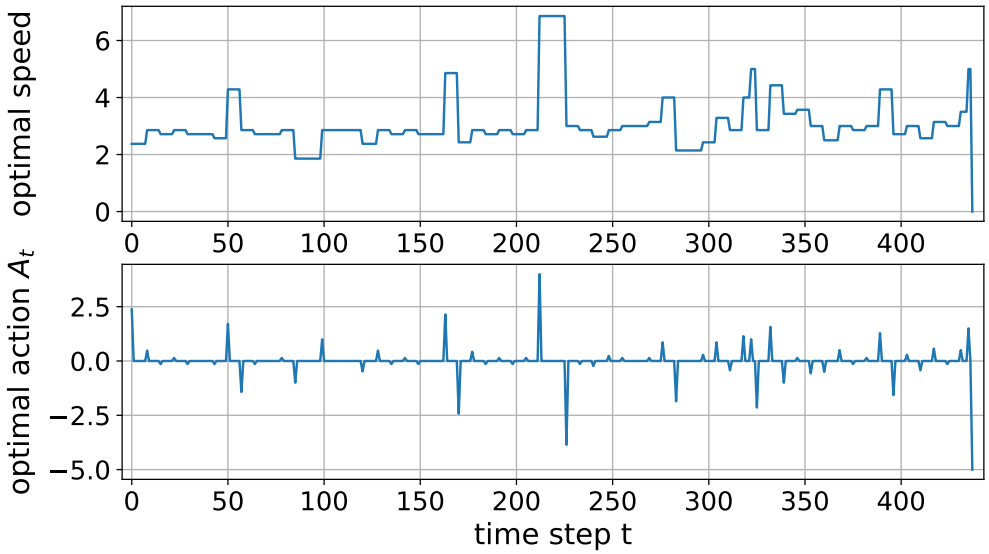


Figure 5.6: Optimal tempo curve and corresponding optimal actions A_t for a continuous agent (piece: J. S. Bach, BWV994). The A_t would be the target values for training an agent with supervised, feed-forward regression.

Method	R_{tue}	R_{on}	$\overline{\lvert d_x \rvert}$	$std(\lvert d_x \rvert)$
	Nottingham (monophonic, 46 test pieces)			
MM-Loc (Chapter 3)	0.43	0.65	3.15	13.15
MM-Loc*(Chapter 3)	0.39	0.59	2.91	12.10
REINFORCE$_{bl}$	0.94	0.96	4.21	4.59
A2C	**0.96**	**0.99**	**2.17**	**3.53**
	MSMD-all (polyphonic, 100 test pieces)			
MM-Loc (Chapter 3)	0.61	0.72	62.34	298.14
MM-Loc*(Chapter 3)	0.52	0.60	**4.28**	**15.76**
REINFORCE$_{bl}$	0.20	0.35	48.61	41.99
A2C	**0.74**	**0.75**	19.25	23.23

Table 5.2: Comparison of score following approaches. Best results are marked in bold. For A2C and REINFORCE$_{bl}$ we report the average over 10 evaluation runs.

completely to the end. In addition, the mean tracking error is lower for A2C and shows a substantially lower standard deviation [4]. The reason is that MM-Loc is formulated as a localization task, predicting a location probability distribution over the score image given the current audio. Musical passages can be highly repetitive, which leads to multiple modes in the location probability distribution, each of which is equally probable. As the MM-Loc tracker follows the mode with highest probability it starts to jump between such ambiguous structures, producing a higher standard deviation for the tracking error and, in the worst case, loses the target. The latter is especially reflected in the low R_{tue} ratios of MM-Loc .

Our MDP formulation of score following addresses this issue, as the agent controls its progression speed for navigating through the sheet image. This restricts the agent as it does not allow for large jumps in the score and, in addition, is much closer to how music is actually performed (e.g. from left to right and top to bottom when excluding repetitions). Our results (especially the ones of A2C) reflect this theoretical advantage.

However, in the case of complex polyphonic scores we also observe that the performance of REINFORCE$_{bl}$ degrades completely. The numbers reported are the outcome of more than five days of training. We already mentioned in Section 5.4 that policy optimization is known to have high variance in the gradient estimate (Greensmith et al., 2004), which is exactly what we observe in our experiments. Even though REINFORCE$_{bl}$ managed to learn a useful policy for the Nottingham dataset it also took more than five days to arrive at that. In contrast, A2C learns a successful policy for the Nottingham dataset in less than six hours and outperforms the baseline method on both datasets. For MSMD it tracks more than 70% of the 100 test pieces entirely to the end without losing the target a single time. This result comes with an average error of only 20 pixels which is about 5mm in a standard A4 page of Western sheet music.

We also report the results of REINFORCE$_{bl}$ to emphasize the potential of RL in this setting. Recall that the underlying MDP is the same for both REINFORCE$_{bl}$ and A2C. The only part that changes is a more powerful learner. All other components including network architecture, optimization algorithm and environment

[4] In the introduction to this chapter I already mentioned that there was a mistake in the original evaluation of the MM-Loc baseline. I corrected this mistake for a re-evaluation in this chapter and updated the results accordingly (marked as MM-Loc*). Initially the evaluation of the baseline was too optimistic and did not detect all *target-lost-events* of the MM-Loc score follower (the tracker continued tracking even though the target position already left its field of view). As a result, the results for the robustness measures R_{tue} and R_{on} were too high (optimistic). On the other hand, these unnoticed *target-lost-events* resulted in large tracking errors, which explains why the mean and standard deviation of the absolute tracking error $|d_x|$ were initially too high. A re-evaluation shows that the MM-Loc is more precises when it does manage to track a piece. However, it fails in this task on 48% of all pieces on MSMD and even 61% on Nottingham (see R_{tue} values).

remain untouched. Considering that deep RL is currently one of the most intensively researched areas in machine learning, we can expect further improvement in the score following task whenever there is an advance in RL itself.

5.6 Conclusion

In this chapter we proposed a formulation of score following in sheet music images as a Markov decision process and showed how to address it with state-of-the-art deep reinforcement learning. Experimental results on monophonic and polyphonic piano music show that this is competitive with related methods addressing the same task (Chapter 3, (Dorfer et al., 2016b)). We would like to close with a discussion of some specific aspects that point to interesting future perspectives.

Firstly, we trained all agents using a continuous reward signal computed by interpolating the target (*ground truth*) location between successive onsets and note heads. Reinforcement learners can, of course, also learn from a *delayed* signal (e.g. non-zero rewards only at actual onsets or even bar lines or downbeats). This further implies that we could, for example, take one of our models trained on the synthesized audios, annotate a set of real performance audios at the bar level (which is perfectly feasible), and then fine-tune the models with the very same algorithms, with the sole difference that for time points without annotation the environment simply returns a neutral reward of zero.

Secondly, we have already started to experiment with continuous control agents that directly predict the required tempo changes, rather than relying on a discrete set of action. Continuous control has proven to be very successful in other domains (Duan et al., 2016) and would allow for a perfect alignment of sheet music and audio (cf. Figure 5.6).

A final remark concerns RL in general. For many RL benchmarks we are given a simulated environment that the agents interact with. These environments are fixed problems without a natural split into training, validation and testing situations. This is different in our setting, and one of the main challenges is to learn agents which generalize to unseen pieces and audio conditions. While techniques such as weight-decay, dropout (Srivastava et al., 2014) or batch-normalization (Ioffe and Szegedy, 2015) have become a standard tool for regularization in supervised learning they are not researched in the context of RL. A broad benchmark of these regularizers in the context of RL would be therefore of high relevance.

We think that all of this makes the score following MDP a promising and in our opinion very exciting playground for further research in both music information retrieval and reinforcement learning.

Part II

Deep Learning on Top of Classical Multivariate Statistics

In this second part of my thesis I propose two methodological extensions to deep learning and neural networks in general, which are both inspired by concepts from classical multivariate statistics. In particular, we revisit Canonical Correlation Analysis (CCA) (Hotelling, 1936) and Linear Discriminant Analysis (LDA) (Fisher, 1936) to reuse and extend their core ideas to allow for a combination with deep neural networks utilized as powerful feature learners. Both proposals can be therefore also interpreted as nonlinear versions of their respective linear ancestors. What additionally connects the two approaches from a machine learning perspective, is the fact that they are both trained on statistics that consider all samples in a given mini-batch for computing the parameter updates of the networks. In the case of CCA these statistics are the covariance and cross-covariance matrices of two groups of random variables (e.g. the latent representations of two different neural networks). For LDA, a discriminative model, we consider special covariance matrices capturing the between- and within-class scatter in the given training data. Training a model on statistics of entire mini-batches is in contrast to objectives such as categorical cross entropy, which are a common choice for optimizing classification neural networks, where each sample is considered individually. This difference will become clearer when describing the two approaches in detail in the respective chapters of this part.

To also provide the link to the proposals presented in the first part of my thesis, I would like to note that the CCA layer described in Chapter 6 was already successfully applied on audio and sheet music in Chapter 4[5]. In fact both lines of research were carried out concurrently and turned out to complement each other in a fruitful way. In the rest of this introduction I give a short intuition on the two methods before describing them in detail in the respective chapters.

Canonical Correlation Analysis (CCA) Layer (Chapter 6). CCA is a method from multivariate statistics measuring the linear dependency (correlation) between two groups of random variables. Concrete instances of the latter can be for example hand-engineered features or the activations of the layers within a neural network (Raghu et al., 2017). In addition to quantifying the correlation between such groups of features, CCA also provides us with two linear transformations that project the observations of both groups into a linear subspace where they are maximally correlated. Given this mathematical foundation, I will revisit the problem of cross-modality retrieval (see Chapter 4) and propose a CCA-based special purpose neural network layer that learns better embedding spaces by analytically computing projections that maximize correlation. Recall that cross-modality retrieval encom-

[5] Actually Chapter 4 could have been included in Part I of this thesis, as it also addresses the problem of cross-modality retrieval. I chose to include it in Part II instead because it describes a more general machine learning solution to cross-modality retrieval via joint embedding space learning also for other tasks than audio – sheet music retrieval.

passes retrieval tasks where the fetched items are of a different type than the search query. The state-of-the-art approach to this problem relies on learning a joint embedding space of the two modalities, where items from either modality are retrieved using nearest-neighbor search. I will show in four different experiments covering the modalities image, audio and text that it is beneficial to utilize the analytical projections of CCA as a final embedding layer instead of learning these projection matrices entirely from scratch (a fully connected layer without nonlinearity).

Deep Linear Discriminant Analysis (Deep LDA) (Chapter 7). LDA operates, in contrast to CCA, only on a single group of features originating from C different classes. Similar to CCA it provides us with a linear combination of features, but in this case projecting into a subspace where observations of different classes are maximally separated. To arrive at this special linear combination we first have to solve LDA's generalized eigenvalue problem computed on the between- and within-class scatter matrices already mentioned above. The projection into this linearly separable space is then inferred from the respective eigenvectors of this eigenvalue problem. The corresponding eigenvalues – and this is the important part for Deep LDA – quantify how well the classes are separated in direction of each eigenvector. We make use of this and reformulate LDA as an optimization target to train neural networks that produce discriminative, linearly separable latent representations useful for calcification tasks such as object recognition. In particular, we put LDA on top of a neural network to maximize the eigenvalues and therefore also class separation of the topmost latent representations of this network.

6 End-to-End Cross-Modality Retrieval with CCA Projections and Pairwise Ranking Loss

Cross-modality retrieval encompasses retrieval tasks where the fetched items are of a different type than the search query, e.g., retrieving pictures relevant to a given text query. The state-of-the-art approach to cross-modality retrieval relies on learning a joint embedding space of the two modalities, where items from either modality are retrieved using nearest-neighbor search (see Chapter 4). In this chapter, I introduce a neural network layer based on Canonical Correlation Analysis (CCA) that learns better embedding spaces by analytically computing projections that maximize correlation. In contrast to previous approaches, the CCA Layer (CCAL) allows us to combine existing objectives for embedding space learning, such as pairwise ranking losses, with the optimal projections of CCA. I show the effectiveness of this approach for cross-modality retrieval on three different scenarios (text-to-image, audio-sheet-music and zero-shot retrieval), surpassing both Deep CCA and a multi-view network using freely learned projections optimized by a pairwise ranking loss, especially when little training data is available.

Note that the CCAL is a general approach to cross-modality retrieval which I already applied in Chapter 4 to one particular problem instance, namely joint embedding space learning for linking audio and sheet music. This chapter provides the technical details of the method and is based on the following publication:

- M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer. End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval (IJMIR)*, 7(2):117–128, 2018d.

Personal Contributions I first had the idea to interpret Canonical Correlation Analysis as a multimodal neural network layer when reading the work of Andrew et al. (2013). I am responsible for developing the layer and carried out the experiments. Jan helped a lot with finding an efficient Theano implementation.

6.1 Introduction

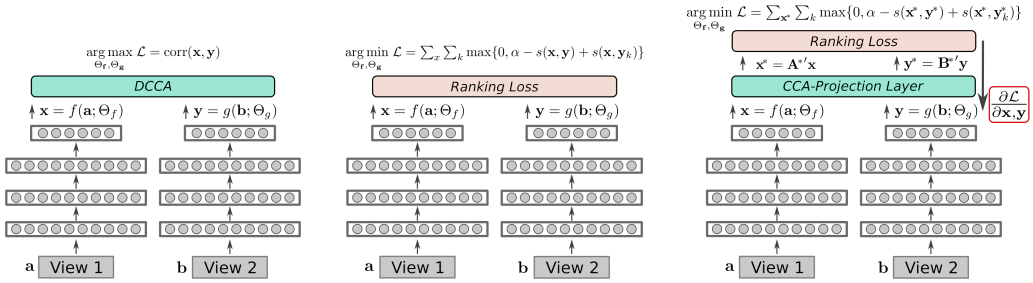
Cross-modality retrieval is the task of retrieving relevant items of a different modality than the search query (e.g., retrieving an image given a text query). One approach to tackle this problem is to define transformations which embed samples from different modalities into a common vector space. We can then project a query into this embedding space, and retrieve, using nearest neighbor search, a corresponding candidate projected from another modality.

A particularly successful class of models uses parametric nonlinear transformations (e.g., neural networks) for the embedding projections, optimized via a retrieval-specific objective such as a pairwise ranking loss (Kiros et al., 2014; Socher et al., 2014). This loss aims at decreasing the distance (a differentiable function such as Euclidean or cosine distance) between matching items, while increasing it between mismatching ones. Specialized extensions of this loss achieved state-of-the-art results in various domains such as natural language processing (Hermann and Blunsom, 2014), image captioning (Karpathy and Fei-Fei, 2015), and text-to-image retrieval (Vendrov et al., 2016).

In a different approach, Yan and Mikolajczyk (2015) propose to learn a joint embedding of text and images using Deep Canonical Correlation Analysis (DCCA) (Andrew et al., 2013). Instead of a pairwise ranking loss, DCCA directly optimizes the correlation of learned latent representations of the two views. Given the correlated embedding representations of the two views, it is possible to perform retrieval via cosine distance. The promising performance of their approach is also in line with the findings of Pereira et al. (2014) who state the following two hypotheses regarding the properties of efficient cross-modal retrieval spaces: First, the embedding spaces should account for low-level cross-modal correlations and second, they should enable semantic abstraction. In (Yan and Mikolajczyk, 2015), both properties are met by a deep neural network — learning abstract representations — that is optimized with DCCA ensuring highly correlated latent representations.

In summary, the optimization of pairwise ranking losses yields embedding spaces that are useful for retrieval, and allows incorporating domain knowledge into the loss function. On the other hand, DCCA is designed to maximize correlation—which has already proven to be useful for cross-modality retrieval (Yan and Mikolajczyk, 2015)—but does not allow to use loss formulations specialized for the task at hand.

In this chapter, we propose a method to combine both approaches in a way that retains their advantages. We develop a *Canonical Correlation Analysis Layer* (CCAL) that can be inserted into a dual-view neural network to produce a maximally correlated embedding space for its latent representations. We can then apply task specific loss functions, in particular the pairwise ranking loss, on the output of this layer. To train a network using the CCA layer we describe how to back-propagate the gradient of this loss function to the dual-view neural network while



(a) DCCA network maximizes correlation via Trace Norm Objective (*TNO*).

(b) Freely-learned embedding projections optimized with ranking loss (*Learned-*$\mathcal{L}_{rank}$).

(c) Canonically correlated projection layer optimized with ranking loss (*CCAL-*$\mathcal{L}_{rank}$).

Figure 6.1: Sketches of cross-modality retrieval networks. The proposed model in (c) unifies (a) and (b) and takes advantage of both: componentwise correlated CCA projections and a pairwise ranking loss for cross-modality embedding space learning. We emphasize that our proposal in (c) requires to backpropagate the ranking loss $\mathcal{L}$ through the analytical computation of the optimally correlated CCA embedding projections $\mathbf{A}^*$ and $\mathbf{B}^*$ (see Equation (6.4)). We thus need to compute their partial derivatives with respect to the network's hidden representations $\mathbf{x}$ and $\mathbf{y}$, i.e. $\frac{\partial \mathbf{A}^*}{\partial \mathbf{x},\mathbf{y}}$ and $\frac{\partial \mathbf{B}^*}{\partial \mathbf{x},\mathbf{y}}$ (addressed in Section 6.4).

relying on automatic differentiation tools such as *Theano* (Bergstra et al., 2010) or *Tensorflow* (Abadi et al., 2016). In our experiments, we show that our proposed method performs better than DCCA and models using pairwise ranking loss alone, especially when little training data is available.

Figure 6.1 compares our proposed approach to the alternatives discussed above. DCCA defines an objective optimizing a dual-view neural network such that its two views will be maximally correlated (Figure 6.1a). Pairwise ranking losses are loss functions to optimize a dual-view neural network such that its two views are well-suited for nearest-neighbor retrieval in the embedding space (Figure 6.1b). In our approach, we boost optimization of a pairwise ranking loss based on cosine distance by placing a special-purpose layer, the CCA projection layer, between a dual-view neural network and the optimization target (Figure 6.1c). Our experiments in Section 6.5 will show the effectiveness of this proposal.

6.2 Canonical Correlation Analysis

In this section we review the concepts of CCA, the basis for our methodology. Let $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$ denote two random column vectors with covariances Σ_{xx} and Σ_{yy} and cross-covariance Σ_{xy}. The objective of CCA is to find two matrices $\mathbf{A}^* \in \mathbb{R}^{d_x \times k}$ and $\mathbf{B}^* \in \mathbb{R}^{d_y \times k}$ composed of k paired column vectors $\mathbf{A}_j$ and $\mathbf{B}_j$ (with $k \leq d_x$ and $k \leq d_y$) that project $\mathbf{x}$ and $\mathbf{y}$ into a common space maximizing their componentwise correlation:

$$(\mathbf{A}^*, \mathbf{B}^*) = \arg\max_{\mathbf{A},\mathbf{B}} \sum_{j=1}^{k} \operatorname{corr}(\mathbf{A}_j'\mathbf{x}, \mathbf{B}_j'\mathbf{y}) \tag{6.1}$$

$$= \arg\max_{\mathbf{A},\mathbf{B}} \sum_{j=1}^{k} \frac{\mathbf{A}_j'\Sigma_{xy}\mathbf{B}_j}{\sqrt{\mathbf{A}_j'\Sigma_{xx}\mathbf{A}_j \; \mathbf{B}_j'\Sigma_{yy}\mathbf{B}_j}} \tag{6.2}$$

Since the objective of CCA is invariant to scaling of the projection matrices, we constrain the projected dimensions to have unit variance. Furthermore, CCA seeks subsequently uncorrelated projection vectors, arriving at the equivalent formulation:

$$(\mathbf{A}^*, \mathbf{B}^*) = \arg\max_{\mathbf{A}'\Sigma_{xx}\mathbf{A}=\mathbf{B}'\Sigma_{yy}\mathbf{B}=\mathbf{I}_k} \operatorname{tr}\left(\mathbf{A}'\Sigma_{xy}\mathbf{B}\right) \tag{6.3}$$

Let $\mathbf{T} = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}$, and let $\mathbf{U}\operatorname{diag}(\mathbf{d})\mathbf{V}'$ be the Singular Value Decomposition (SVD) of $\mathbf{T}$ with ordered singular values $d_i \geq d_{i+1}$. As shown by Mardia et al. (1979), we obtain $\mathbf{A}^*$ and $\mathbf{B}^*$ from the top k left- and right-singular vectors of $\mathbf{T}$:

$$\mathbf{A}^* = \Sigma_{xx}^{-1/2}\mathbf{U}_{:k} \qquad \mathbf{B}^* = \Sigma_{yy}^{-1/2}\mathbf{V}_{:k} \tag{6.4}$$

Moreover, the correlation in the projection space is the sum of the top k singular values:[1]

$$\operatorname{corr}(\mathbf{A}^{*\prime}\mathbf{x}, \mathbf{B}^{*\prime}\mathbf{y}) = \sum_{i \leq k} d_i \tag{6.5}$$

In practice, the covariances and cross-covariance of $\mathbf{x}$ and $\mathbf{y}$ are usually not known, but estimated from a training set of m paired vectors, expressed as matrices $\mathbf{X} \in \mathbb{R}^{d_x \times m}$, $\mathbf{Y} \in \mathbb{R}^{d_y \times m}$ by:

$$\hat{\Sigma}_{xx} = \frac{1}{m-1}\overline{\mathbf{X}}\overline{\mathbf{X}}' + r\mathbf{I} \;\text{ and }\; \hat{\Sigma}_{xy} = \frac{1}{m-1}\overline{\mathbf{X}}\overline{\mathbf{Y}}'. \tag{6.6}$$

$\overline{\mathbf{X}}$ is the centered version of $\mathbf{X}$. $\hat{\Sigma}_{yy}$ is defined analogously to $\hat{\Sigma}_{xx}$. Additionally, we apply a regularization parameter $r\mathbf{I}$ to ensure that the covariance matrices are positive definite. Substituting these estimates for Σ_{xx}, Σ_{xy} and Σ_{yy}, respectively, we can compute $\mathbf{A}^*$ and $\mathbf{B}^*$ using Equation (6.4).

[1] We understand the correlation of two vectors to be defined as $\operatorname{corr}(\mathbf{x}, \mathbf{y}) = \sum_i \sum_j \operatorname{corr}(x_i, y_j)$.

6.3 Cross-Modality Retrieval Baselines

In this section we review the two most related works forming the basis for our approach.

6.3.1 Deep Canonical Correlation Analysis

Andrew et al. (2013) propose an extension of CCA to learn parametric nonlinear transformations of two random vectors, such that their correlation is maximized. Let $\mathbf{a} \in \mathbb{R}^{d_a}$ and $\mathbf{b} \in \mathbb{R}^{d_b}$ denote two random vectors, and let $\mathbf{x} = f(\mathbf{a}; \Theta_f)$ and $\mathbf{y} = g(\mathbf{b}; \Theta_g)$ denote their nonlinear transformations, parameterized by Θ_f and Θ_g. DCCA optimizes the parameters Θ_f and Θ_g to maximize the correlation of the topmost hidden representations $\mathbf{x}$ and $\mathbf{y}$. For $d_x = d_y = k$, this objective corresponds to Equation 6.5, i.e., the sum of all singular values of $\mathbf{T}$, also called the trace norm:

$$\text{corr}(\mathbf{A}^{*\prime} f(\mathbf{a}; \Theta_f), \mathbf{B}^{*\prime} g(\mathbf{b}; \Theta_g)) = ||\mathbf{T}||_{\text{tr}}. \tag{6.7}$$

Andrew et al. (2013) show how to compute the gradient of this *Trace Norm Objective* (TNO) with respect to $\mathbf{x}$ and $\mathbf{y}$. Assuming f and g are differentiable with respect to Θ_f and Θ_g (as is the case for neural networks), this allows to optimize the nonlinear transformations via gradient-based methods.

Yan and Mikolajczyk (2015) suggest the following procedure to utilize DCCA for cross-modality retrieval: first, neural networks f and g are trained using the TNO, with $\mathbf{a}$ and $\mathbf{b}$ representing different views of an entity (e.g. image and text); then, after the training is finished, the CCA projections are computed using Equation (6.4), and all retrieval candidates are projected into the embedding space; finally, at test time, queries of either modality are projected into the embedding space, and the best-matching sample from the other modality is found through nearest neighbor search using the cosine distance. Figure 6.2 provides a summary of the entire retrieval pipeline. In our experiments, we will refer to this approach as *DCCA-2015*.

DCCA is limited by design to use the objective function described in Equation (6.7), and only seeks to maximize the correlation in the embedding space. During training, the CCA projection matrices are never computed, nor are the samples projected into the common retrieval space. All the retrieval steps—most importantly, the computation of CCA projections—are performed only once after the networks f and g have been optimized. This restricts potential applications, because we cannot use the projected data as an input to subsequent layers or task-specific objectives. We will show how our approach overcomes this limitation in Section 6.4.

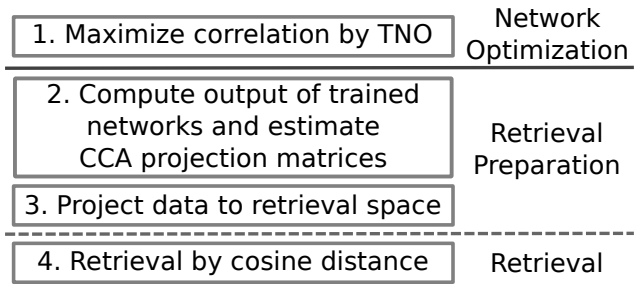


Figure 6.2: *DCCA* retrieval pipeline proposed in (Yan and Mikolajczyk, 2015). Note that all processing steps below the solid line are performed after network optimization is complete.

6.3.2 Pairwise Ranking Loss

Kiros et al. (2014) learn a multimodal joint embedding space for images and text. They use the cosine of the angle between two corresponding vectors $\mathbf{x}$ and $\mathbf{y}$ as a scoring function, i.e., $s(\mathbf{x}, \mathbf{y}) = \cos(\mathbf{x}, \mathbf{y})$. Then, they optimize a pairwise ranking loss

$$\mathcal{L}_{rank} = \sum_{\mathbf{x}} \sum_{k} \max\{0, \alpha - s(\mathbf{x}, \mathbf{y}) + s(\mathbf{x}, \mathbf{y}_k)\} \tag{6.8}$$

where $\mathbf{x}$ is an embedded sample of the first modality, $\mathbf{y}$ is the matching embedded sample of the second modality, and $\mathbf{y}_k$ are the contrastive (mismatching) embedded samples of the second modality (in practice, all mismatching samples in the current mini-batch). The hyper-parameter α defines the margin of the loss function. This loss encourages an embedding space where the cosine distance between matching samples is lower than the cosine distance of mismatching samples.

In this setting, the networks f and g have to learn the embedding projections freely from randomly initialized weights. Since the projections are learned from scratch by optimizing a ranking loss, in our experiments we denote this approach by *Learned-*$\mathcal{L}_{rank}$. Figure 6.1b shows a sketch of this paradigm.

6.4 Learning with Canonically Correlated Embedding Projections

In the following we explain how to bring both concepts — DCCA and Pairwise Ranking Losses — together to enhance cross-modality embedding space learning.

6.4.1 Motivation

We start by providing an intuition on why we expect this combination to be fruitful: *DCCA-2015* maximizes the correlation between the latent representations of two different neural networks via the TNO derived from classic CCA. As correlation and cosine distance are related, we can also use such a network for cross-modality retrieval (Yan and Mikolajczyk, 2015). Kiros et al. (2014), on the other hand, learn a cross-modality retrieval embedding by optimizing an objective customized for the task at hand. The motivation for our approach is that we want to benefit from both: a task specific retrieval objective, and componentwise optimally correlated embedding projections.

To achieve this, we devise a *CCA layer* that analytically computes the CCA projections $\mathbf{A}^*$ and $\mathbf{B}^*$ during training, and projects incoming samples into the embedding space. The projected samples can then be used in subsequent layers, or for computing task-specific losses such as the pairwise ranking loss. Figure 6.1c illustrates the central idea of our combined approach. Compared to Figure 6.1b, we insert an additional linear transformation. However, this transformation is not learned (otherwise it could be merged with the previous layer, which is not followed by a nonlinearity). Instead, it is computed to be the transformation that maximizes componentwise correlation between the two views. $\mathbf{A}^*$ and $\mathbf{B}^*$ in Figure 6.1c are the very projections given by Equation (6.4) in Section 6.2.

In theory, optimizing a pairwise ranking loss alone could yield projections equivalent to the ones computed by CCA. In practice, however, we observe that the proposed combination gives much better cross-modality retrieval results (see Section 6.5).

Our design requires backpropagating errors through the analytical computation of the CCA projection matrices. DCCA (Andrew et al., 2013) does not cover this, since projecting the data is not necessary for optimizing the TNO. In the remainder of this section, we discuss how to establish gradient flow (backpropagation) through CCA's optimal projection matrices. In particular, we require the partial derivatives $\frac{\partial \mathbf{A}^*}{\partial \mathbf{x},\mathbf{y}}$ and $\frac{\partial \mathbf{B}^*}{\partial \mathbf{x},\mathbf{y}}$ of the projections with respect to their input representations $\mathbf{x}$ and $\mathbf{y}$. This will allow us to use CCA as a layer within a multimodality neural network, instead of as a final objective (TNO) for correlation maximization only.

6.4.2 Gradient of CCA Projections

As mentioned above, we can compute the canonical correlation along with the optimal projection matrices from the singular value decomposition $\mathbf{T} = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2} = \mathbf{U}\,\mathrm{diag}(\mathbf{d})\mathbf{V}'$. Specifically, we obtain the correlation as $\mathrm{corr}(\mathbf{A}^{*\prime}\mathbf{x}, \mathbf{B}^{*\prime}\mathbf{y}) = \sum_i d_i$, and the projections as $\mathbf{A}^* = \Sigma_{xx}^{-1/2}\mathbf{U}$ and $\mathbf{B}^* = \Sigma_{yy}^{-1/2}\mathbf{V}$. For DCCA, it suffices to compute the gradient of the total correlation wrt. $\mathbf{x}$ and $\mathbf{y}$ in order to backpropagate

it through the two networks f and g. Using the chain rule, Andrew et al. (2013) decompose this into the gradients of the total correlation wrt. Σ_{xx}, Σ_{xy} and Σ_{yy}, and the gradients of those wrt. $\mathbf{x}$ and $\mathbf{y}$. Their derivations of the former make use of the fact that both the gradient of $\sum_i d_i$ wrt. $\mathbf{T}$ and the gradient of $||\mathbf{T}||_{\text{tr}}$ (the trace norm objective in Equation (6.7)) wrt. $\mathbf{T}'\mathbf{T}$ have a simple form; see Section 7 in (Andrew et al., 2013) for details.

In our case where we would like to backpropagate errors through the CCA transformations, we instead need the gradients of the projected data $\mathbf{x}^* = \mathbf{A}^{*\prime}\mathbf{x}$ and $\mathbf{y}^* = \mathbf{B}^{*\prime}\mathbf{y}$ wrt. $\mathbf{x}$ and $\mathbf{y}$, which requires the partial derivatives $\frac{\partial \mathbf{A}^*}{\partial \mathbf{x},\mathbf{y}}$ and $\frac{\partial \mathbf{B}^*}{\partial \mathbf{x},\mathbf{y}}$. We could again decompose this into the gradients wrt. $\mathbf{T}$, the gradients of $\mathbf{T}$ wrt. Σ_{xx}, Σ_{xy} and Σ_{yy} and the gradients of those wrt. $\mathbf{x}$ and $\mathbf{y}$. However, while the gradients of $\mathbf{U}$ and $\mathbf{V}$ wrt. $\mathbf{T}$ are known (Papadopoulo and Lourakis, 2000), they involve solving $O((d_x d_y)^2)$ linear 2×2 systems. Instead, we reformulate the solution to use two symmetric eigendecompositions $\mathbf{T}\mathbf{T}' = \mathbf{U}\,\text{diag}(\mathbf{e})\mathbf{U}'$ and $\mathbf{T}'\mathbf{T} = \mathbf{V}\,\text{diag}(\mathbf{e})\mathbf{V}'$ (Equation 270 in (Petersen and Pedersen, 2012)). This gives us the same left and right eigenvectors we would obtain from the SVD, along with the squared singular values ($e_i = d_i^2$). The gradients of eigenvectors of symmetric real eigensystems have a simple form (Magnus, 1985) and both $\mathbf{T}\mathbf{T}'$ and $\mathbf{T}'\mathbf{T}$ are differentiable wrt. $\mathbf{x}$ and $\mathbf{y}$.

To summarize: in order to obtain an efficiently computable definition of the gradient for CCA projections, we have reformulated the forward pass (the computation of the CCA transformations). Our formulation using two eigen-decompositions translates into a series of computation steps that are differentiable in a graph-based, auto-differentiating math compiler such as *Theano* (Bergstra et al., 2010), which, together with the chain rule, gives an efficient implementation of the CCA layer gradient for training our network[2]. For a detailed description of the CCA layer forward pass we refer to Algorithm 1 in the Appendix of this chapter. As the technical implementation is not straight-forward, we also discuss the crucial steps in the Appendix.

Thus, we now have the means to benefit from the optimal CCA projections but still optimize for a task-specific objective. In particular, we utilize the *pairwise ranking loss* of Equation (6.8) on top of an intermediate CCA embedding projection layer. We denote the proposed retrieval network of Figure 6.1c as *CCAL*-$\mathcal{L}_{rank}$ in our experiments (*CCAL* refers to CCA Layer).

[2]The code of our implementation of the CCA layer is available at `https://github.com/CPJKU/cca_layer`.

6.5 Experiments

We evaluate our approach (*CCAL-*$\mathcal{L}_{rank}$) in cross-modality retrieval experiments on two image-to-text and one audio-to-sheet-music dataset. Additionally, we provide results on two zero-shot text-to-image retrieval scenarios proposed in (Reed et al., 2016). For comparison, we consider the approach of Yan and Mikolajczyk (2015) (*DCCA-2015*), our own implementation of the TNO (denoted by *DCCA*), as well as the freely learned projection embeddings (*Learned-*$\mathcal{L}_{rank}$) optimizing the ranking loss of Kiros et al. (2014).

The task for all three datasets is to retrieve the correct counterpart when given an instance of the other modality as a search query (see also Chapter 4 addressing the same task in the more specific setting of audio and sheet music). For retrieval, we use the cosine distance in embedding space for all approaches. First, we embed all candidate samples of the target modality into the retrieval embedding space. Then, we embed the query element $\mathbf{y}$ with the second network and select its nearest neighbor $\mathbf{x}_j$ of the target modality. Figure 6.3 shows a sketch of this retrieval by embedding space learning paradigm.

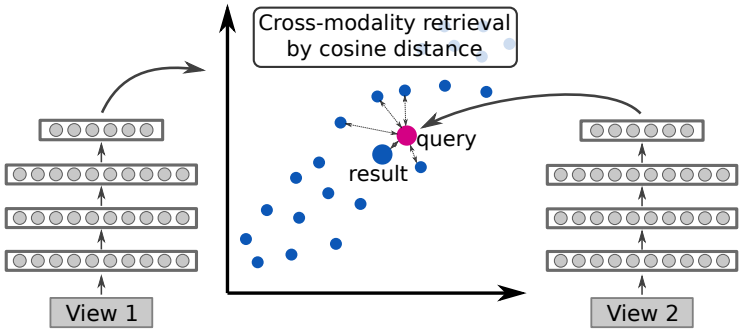


Figure 6.3: Sketch of cross-modality retrieval. The blue dots are the embedded candidate samples. The red dot is the embedding of the search query. The larger blue dot highlights the closest candidate selected as the retrieval result.

As evaluation measures, we consider the *Recall@k (R@k in %)* as well as the *Median Rank (MR)* and the *Mean Reciprocal Rank (MRR in %)*. The *R@k* rate (higher is better) is the ratio of queries which have the correct corresponding counterpart in the first k retrieval results. The *MR* (lower is better) is the median position of the target in a similarity-ordered list of available candidates. Finally, we define the *MRR* (higher is better) as the mean value of $1/rank$ over all queries where *rank* is again the position of the target in the similarity ordered list of available candidates.

	A man in a white cowboy hat reclines in front of a window in an airport. A young man rests on an airport seat with a cowboy hat over his face. A woman relaxes on a couch , with a white cowboy hat over her head. A man is sleeping inside on a bench with his hat over his eyes. A person is sleeping at an airport with a hat on their head.
	a green and brown embankment with brown houses on the right and a light brown sandy beach at the dark blue sea on the left; a dark mountain range behind it and white clouds in a light blue sky in the background;

Table 6.1: Example images for Flickr30k (top) and IAPR TC-12 (bottom)

6.5.1 Image-Text Retrieval

In the first part of our experiments, we consider *Flickr30k* and *IAPR TC-12*, two publicly available datasets for image-text cross-modality retrieval. Flickr30k consists of image-caption pairs, where each image is annotated with five different textual descriptions. The train-validation-test split for Flickr30k is 28000-1000-1000. In terms of evaluation setup, we follow *Protocol 3* of (Yan and Mikolajczyk, 2015) and concatenate the five available captions into one, meaning that only one, but richer text annotation remains per image. This is done for all three sets of the split. The second image-text dataset, IAPR TC-12, contains 20000 natural images where only one—but compared to Flickr30k more detailed—caption is available for each image. As no predefined train-validation-test split is provided, we randomly select 1000 images for validation and 2000 for testing, and keep the rest for training. Yan and Mikolajczyk (2015) also use 2000 images for testing, but did not explicitly mention holdout images for validation. Table 6.1 shows an example image along with its corresponding captions or caption for either dataset.

The input to our networks is a 4096-dimensional image feature vector along with a corresponding text vector representation which has dimensionality 5793 for Flickr30k and 2048 for IAPR TC-12. The image embedding is computed from the last hidden layer of a network pretrained on ImageNet (Deng et al., 2009) (layer *fc7* of *CNN_S* by Chatfield et al. (2014)). In terms of text pre-processing, we follow

Table 6.2: Retrieval results on IAPR TC-12. "DCCA-2015" is taken from (Yan and Mikolajczyk, 2015).

	Image-to-Text					Text-to-Image				
Method	R@1	R@5	R@10	MR	MRR	R@1	R@5	R@10	MR	MRR
DCCA-2015	30.2	57.0	-	-	42.6	29.5	60.0	-	-	41.5
DCCA	31.0	58.7	70.4	3.6	43.9	29.5	58.2	70.5	4.0	42.7
Learned-$\mathcal{L}_{rank}$	22.3	50.7	63.8	5.2	35.7	21.6	50.1	63.3	5.5	35.1
CCAL-$\mathcal{L}_{rank}$	31.6	61.0	72.2	3.0	45.0	29.6	60.0	72.2	3.6	43.5

Yan and Mikolajczyk (2015), tokenizing and lemmatizing the raw captions as the first step. Based on the lemmatized captions, we compute $l2$-normalized TF/IDF-vectors, omitting words with an overall occurrence smaller than five for Flickr30k and three for IAPR TC-12, respectively. The image representation is processed by a linear dense layer with 128 units, which will also be the dimensionality k of the resulting retrieval embedding. The text vector is fed through two batch-normalized (Ioffe and Szegedy, 2015) dense layers of 1024 units each and the ELU activation function (Clevert et al., 2016). As a last layer for the text representation network, we again apply a dense layer with 128 linear units.

For a fair comparison, we keep the structure and number of parameters of all networks in our experiments the same. The only difference between the networks are the objectives and the hyper-parameters used for optimization. Optimization is performed using Stochastic Gradient Descent (SGD) with the *adam* update rule (Kingma and Ba, 2015) (for details please see the appendix of this chapter).

Table 6.2 lists our results on IAPR TC-12. Along with our experiments, we also show the results reported in (Yan and Mikolajczyk, 2015) as a reference (*DCCA-2015*). However, a direct comparison to our results may not be fair: *DCCA-2015* uses a different ImageNet-pretrained network for the image representation, and finetunes this network while we keep it fixed. This is because our interest is in comparing the methods in a stable setting, not in obtaining the best possible results. Our implementation of the TNO (*DCCA*) uses the same objective as *DCCA-2015*, but is trained using the same network architecture as our remaining models and permits a direct comparison. Additionally, we repeat each of the experiments 10 times with different initializations and report the mean for each of the evaluation measures.

When taking a closer look at Table 6.2, we observe that our results achieved by optimizing the TNO (*DCCA*) surpass the results reported in (Yan and Mikolajczyk, 2015). We already discussed above that the two versions are not directly comparable. However, given this result, we consider our implementation of *DCCA* as a valid baseline for our experiments in Section 6.5.2 where no results are available in

Table 6.3: Retrieval results on Flickr30k. "DCCA-2015" is taken from (Yan and Mikolajczyk, 2015).

	Image-to-Text					Text-to-Image				
Method	R@1	R@5	R@10	MR	MRR	R@1	R@5	R@10	MR	MRR
DCCA-2015	27.9	56.9	68.2	4	-	26.8	52.9	66.9	4	-
DCCA	31.6	59.2	69.3	3.3	44.2	30.3	58.3	69.2	3.8	43.1
Learned-$\mathcal{L}_{rank}$	23.7	50.5	63.0	5.3	36.3	23.6	51.0	62.5	5.2	36.5
CCAL-$\mathcal{L}_{rank}$	32.0	59.2	70.4	3.2	44.8	29.9	58.8	70.2	3.7	43.3

the literature. When looking at the performance of *CCAL-*$\mathcal{L}_{rank}$ we further observe that it outperforms all other methods, although the difference to *DCCA* is not pronounced for all of the measures. Comparing *CCAL-*$\mathcal{L}_{rank}$ with the freely-learned projection matrices (*Learned-*$\mathcal{L}_{rank}$) we observe a much larger performance gap. This is interesting, as in principle the learned projections could converge to exactly the same solution as *CCAL-*$\mathcal{L}_{rank}$. We take this as a quantitative confirmation that the learning process benefits from CCA's optimal projection matrices.

In Table 6.3, we list our results on the Flickr30k dataset. As above, we show the retrieval performances reported in (Yan and Mikolajczyk, 2015) as a baseline along with our results and observe similar behavior as on IAPR TC-12. Again, we point out the poor performance of the freely-learned projections (*Learned-*$\mathcal{L}_{rank}$) in this experiment. Keeping this observation in mind, we will notice a different behavior in the experiments in Section 6.5.2.

Note that there are various other methods reporting results on Flickr30k (Karpathy et al., 2014; Socher et al., 2014; Mao et al., 2014; Kiros et al., 2014) which partly surpass ours, for example by using more elaborate processing of the textual descriptions or more powerful ImageNet models. We omit these results as we focus on the comparison of *DCCA* and freely-learned projections with the proposed CCA projection embedding layer.

6.5.2 Audio-Sheet-Music Retrieval

For the second set of experiments, we consider the Nottingham piano midi dataset (Boulanger-lewandowski et al., 2012). The dataset is a collection of midi files split into train, validation and test set which we already used in Dorfer et al. (2016b) for experiments on end-to-end score-following in sheet-music images (see Chapter 3). Here, we tackle the problem of audio-sheet-music retrieval, i.e. matching short snippets of music (audio) to corresponding parts in the sheet music (image). Figure 6.4 shows examples of such correspondences.

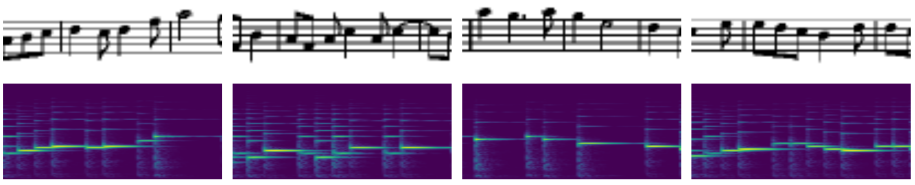


Figure 6.4: Example of the data considered for audio-sheet-music (image) retrieval. Top: short snippets of sheet music images. Bottom: Spectrogram excerpts of the corresponding music audio.

We conduct this experiment for two reasons[3]: First, to show the advantage of the proposed method over different domains. Second, the data and application is of high practical relevance in the domain of Music Information Retrieval (MIR). A system capable of linking sheet music (images) and the corresponding music (audio) would be useful in many content-based musical retrieval scenarios.

In terms of audio preparation, we compute log frequency spectrograms with a sample rate of 22.05kHz, a FFT window size of 2048, and a computation rate of 31.25 frames per second. These spectrograms (136 frequency bins) are then directly fed into the audio part of the cross-modality networks. Figure 6.4 shows a set of audio-to-sheet correspondences presented to our network for training. One audio excerpt comprises 100 frames and the dimension of the sheet image snippet is 40×100 pixels. Overall this results in 270,705 train, 18,046 validation and 16,042 test audio-sheet-music pairs. This is an order of magnitude more training data than for the image-to-text datasets of the previous section.

In the experiments in Section 6.5.1, we relied on pre-trained ImageNet features and relatively shallow fully connected text-feature processing networks. The model here differs from this, as it consists of two deep convolutional networks learned entirely from scratch. Our architecture is a VGG-style (Simonyan and Zisserman, 2015) network consisting of sequences of 3×3 convolution stacks followed by 2×2 max pooling. To reduce the dimensionality to the desired correlation space dimensionality k (in this case 32) we insert as a final building block a 1×1 convolution having k feature maps followed by global average pooling (Lin et al., 2014) (for further architectural details we again refer to the appendix of this manuscript).

Table 6.4 lists our result on audio-to-sheet music retrieval. As in the experiments on images and text, the proposed CCA projection embedding layer trained with

[3] I already elaborated on this extensively in Chapter 4. However, as the experiments here are the first ones where I performed retrieval experiments on audio and sheet music utilizing the CCA layer I kept them in this thesis to provide a complete picture of how the method evolved. The experiments described in Chapter 4 were carried out later and deal with more complex musical material.

Table 6.4: Retrieval results on Nottingham dataset (Audio-to-Sheet-Music Retrieval).

	Sheet-to-Audio					Audio-to-Sheet				
Method	R@1	R@5	R@10	MR	MRR	R@1	R@5	R@10	MR	MRR
DCCA	42.0	88.2	93.3	2	62.2	44.6	87.9	93.2	2	63.5
Learned-$\mathcal{L}_{rank}$	40.7	89.6	95.6	2	61.7	41.4	88.9	95.4	2	61.9
CCAL-$\mathcal{L}_{rank}$	44.1	93.3	97.7	2	65.3	44.5	91.6	96.7	2	64.9

pairwise ranking loss outperforms the other models. Recalling the results from Section 6.5.1, we observe an increased performance of the freely-learned embedding projections. On measures such as R@5 or R@10 it achieves similar to or better performance than *DCCA*. One of the reasons for this could be the fact that there is an order of magnitude more training data available for this task to learn the projection embedding from random initialization. Still, our proposed combination of both concepts (*CCAL-*$\mathcal{L}_{rank}$) achieves highest retrieval scores.

6.5.3 Performance in Small Data Regime

The above results suggest that the benefit of using a CCA projection layer (*CCAL-*$\mathcal{L}_{rank}$) over a freely-learned projection becomes most evident when few training data is available. To examine this assumption, we repeat the audio-to-sheet-music experiment of the previous section, but use only 10% of the original training data ($\approx$ 27000 samples). We stress the fact that the learned embedding projection of *Learned-*$\mathcal{L}_{rank}$ could converge to exactly the same solution as the CCA projections of *CCAL-*$\mathcal{L}_{rank}$. Table 6.5 summarizes the low data regime results for the three methods. Consistent with our hypothesis, we observe a larger gap between *Learned-*$\mathcal{L}_{rank}$ and *CCAL-*$\mathcal{L}_{rank}$ compared to the one obtained with all training data in Table 6.4. We conclude that a network might be able to learn suitable embedding projections when sufficient training data is available. However, when having fewer training samples, the proposed CCA projection layer strongly supports embedding space learning. In addition, we also looked into the retrieval performance of *Learned-*$\mathcal{L}_{rank}$ and *CCAL-*$\mathcal{L}_{rank}$ on the training set and observe comparable performance. This indicates that the CCA layer also acts as a regularizer and helps to generalize to unseen samples.

6.5.4 Zero-Shot Image-Text Retrieval

Our last set of experiments focuses on a slightly modified retrieval setting, namely image-text *zero-shot retrieval* (Reed et al., 2016). Given a set of image-text pairs originating from C different categories the data is split into a class-disjoint training,

Table 6.5: Retrieval results on audio-to-sheet-music retrieval when using only 10% of the train data.

	Sheet-to-Audio					Audio-to-Sheet				
Method	R@1	R@5	R@10	MR	MRR	R@1	R@5	R@10	MR	MRR
DCCA	20.0	53.6	65.4	5	35.3	22.7	54.7	65.8	4	37.3
Learned-$\mathcal{L}_{rank}$	11.3	35.2	47.6	12	23.0	12.6	35.2	47.2	12	23.7
CCAL-$\mathcal{L}_{rank}$	22.2	59.2	70.7	4	38.8	25.0	59.3	70.9	4	40.4

Figure 6.5: Example images of CUB-200 birds and Oxford Flowers along with textual descriptions collected by Reed et al. (2016) for zero-shot retrieval from text.

validation and test sets having no categorical overlap. This implies that at test time we aim to retrieve images from textual queries describing categories (semantic concepts) never seen before, neither for training, nor for validation.

Reed et al. (2016) collected and provided textual descriptions for two publicly available datasets, the *CUB-200 bird image dataset* (Welinder et al., 2010) and the *Oxford Flowers dataset* (Nilsback and Zisserman, 2008). According to the definition of zero-shot retrieval above we follow Reed et al. (2016) and split CUB into 100 train, 50 validation and 50 test categories. Flowers is split into 82 train and 20 validation / test classes respectively. Figure 6.5 shows some example images along with their textual descriptions.

Besides the modified, harder retrieval setting there is a second difference to the text-image retrieval experiments carried out in Section 6.5.1. Instead of using hand engineered textual features (e.g. TF-IDF) or unsupervised textual feature learning (e.g. word2vec (Mikolov et al., 2013)) the authors in (Reed et al., 2016) employ Convolutional Recurrent Neural Networks (CRNN) to learn the latent text representations directly from the raw descriptions. In particular, they feed the descriptions as one-hot-word encodings to the text processing part of their networks. In terms of image representations they still rely on 1024-dimensional pretrained Ima-

Table 6.6: Zero-shot retrieval results on Cub and Flowers.

Method	Flowers	Birds
Attributes (Reed et al., 2016)	-	50.0
Word2Vec (Reed et al., 2016)	52.1	33.5
Word CNN (Reed et al., 2016)	56.3	43.3
Word CNN-RNN (Reed et al., 2016)	59.6	48.7
Word CNN + CCAL	62.2	52.2
Word CNN-RNN + CCAL	64.0	49.8

geNet features. The feature learning part and the network architectures used for our experiments follows exactly the descriptions provided in (Reed et al., 2016). The sole difference is, that we again replace the topmost embedding layer with the proposed CCA projection layer in combination with a pairwise ranking loss.

Table 6.6 compares the retrieval results of the respective methods on the two zero-shot retrieval datasets. To allow for a direct comparison with the results reported in (Reed et al., 2016) we follow their evaluation setup and report the *Average Precision (AP@50)*. The AP@50 is the percentage of the top-50 scoring images whose class matches that of the text query, averaged over the 50 test classes. In (Reed et al., 2016) the best retrieval performance for both datasets (when considering only feature learning) is achieved by having a CRNN directly processing the textual descriptions. What is also interesting is the substantial performance gain with respect to unsupervised word2vec features.

For the Birds dataset, as an alternative to the textual descriptions, there are manually created fine-grained attributes available for each of the images. When relying on these attributes Reed et al. (2016) report state of the art results on the dataset not reached by their text processing neural networks.

In the bottom part of Table 6.6, we report the performance of the same architectures optimized using our proposed CCA layer in combination with a pairwise ranking loss. We observe that the CCA layer is able to improve the performance of both models on both datasets. The gain in retrieval performance within a model class is largest for the convolution only (CNN) text processing models ($\approx$ 9 percentage points for the Flowers dataset and $\approx$ 6 for CUB). For the birds dataset the *Word CNN + CCAL* even outperforms the models relying on manually encoded attributes by achieving an AP@50 of 52.2.

6.6 Conclusion

We have shown how to use the optimal projection matrices of CCA as the weights of an embedding layer within a multi-view neural network. With this CCA layer,

it becomes possible to optimize for a specialized loss function (e.g., related to a retrieval task) on top of this, exploiting the correlation properties of a latent space provided by CCA. As this requires to establish gradient flow through CCA, we formulate it to allow easy computation of the partial derivatives $\frac{\partial \mathbf{A}^*}{\partial \mathbf{x},\mathbf{y}}$ and $\frac{\partial \mathbf{B}^*}{\partial \mathbf{x},\mathbf{y}}$ of CCA's projection matrices $\mathbf{A}^*$ and $\mathbf{B}^*$ with respect to the input data $\mathbf{x}$ and $\mathbf{y}$. With this formulation, we can incorporate CCA as a building block within multimodality neural networks that produces maximally-correlated projections of its inputs. In our experiments, we use this building block within a cross-modality retrieval setting, optimizing a network to minimize a cosine distance based pairwise ranking loss of the componentwise-correlated CCA projections. Experimental results show that when using the cosine distance for retrieval (as is common for correlated views), this is superior to optimizing a network for maximally-correlated projections (as done in DCCA), or not using CCA at all. This observation holds in our experiments on a variety of different modality pairs as well as two different retrieval scenarios.

When investigating the experimental results in more detail, we find that the correlation-based methods (DCCA, CCAL) consistently outperform the models that learn the embedding projections from scratch. A direct comparison of DCCA with the proposed CCAL-$\mathcal{L}_{rank}$ reveals two learning scenarios where CCAL-$\mathcal{L}_{rank}$ is superior: (1) the low data regime, where we found that the CCA layer acts as a strong regularizer to prevent over-fitting; (2) when learning the entire retrieval representation (network parameterization) from scratch, not relying on pre-trained or hand-crafted features (see Section 6.5.2). Our intuition on this is that incorporating the task-specific retrieval objective already during training encourages the networks to learn embedding representations that are beneficial for retrieval at test-time. This is the important conceptual difference compared to the Trace Norm Objective (TNO) of DCCA, which does not focus on the retrieval task. However, when using the CCA layer we also inherit one drawback of the pairwise ranking loss, which is the additional hyper-parameter (margin α) that needs to be determined on the validation set.

Finally, we would like to emphasize that our CCA layer is a general network component which could provide a useful basis for further research, e.g., as an intermediate processing step for learning binary cross-modality retrieval representations.

6.7 Appendix

Implementation Details

Backpropagating the errors through the CCA projection matrices is not trivial. The optimal CCA projection matrices are given by $\mathbf{A}^* = \Sigma_{xx}^{-1/2}\mathbf{U}$ and $\mathbf{B}^* = \Sigma_{yy}^{-1/2}\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are derived from the singular value decomposition of $\mathbf{T} = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2} = \mathbf{U}\,\mathrm{diag}(\mathbf{d})\mathbf{V}'$ (see Section 6.2). The proposed model needs to backpropagate the errors through the CCA transformations, i.e., it requires the gradients of the projected data $\mathbf{x}^* = \mathbf{A}^{*\prime}\mathbf{x}$ and $\mathbf{y}^* = \mathbf{B}^{*\prime}\mathbf{y}$ wrt. $\mathbf{x}$ and $\mathbf{y}$. Applying the chain rule, this further requires the gradients of $\mathbf{U}$ and $\mathbf{V}$ wrt. $\mathbf{T}$, and the gradients of $\mathbf{T}$, $\Sigma_{xx}^{-1/2}$, Σ_{xy} and $\Sigma_{yy}^{-1/2}$ wrt. $\mathbf{x}$ and $\mathbf{y}$.

The main technical challenge is that common auto-differentiation tools such as Theano (Bergstra et al., 2010) or Tensor Flow (Abadi et al., 2016) do not provide derivatives for the inverse squared root and singular value decomposition of a matrix.[4] To overcome this, we replace the inverse squared root of a matrix by using its Cholesky decomposition as described in (Hardoon et al., 2004). Furthermore, we note that the singular value decomposition is required to obtain the matrices $\mathbf{U}$ and $\mathbf{V}$, but in fact those matrices can alternatively be obtained by solving the eigendecomposition of $\mathbf{T}\mathbf{T}' = \mathbf{U}\,\mathrm{diag}(\mathbf{e})\mathbf{U}'$ and $\mathbf{T}'\mathbf{T} = \mathbf{V}\,\mathrm{diag}(\mathbf{e})\mathbf{V}'$ (Petersen and Pedersen, 2012, Eq. 270). This yields the same left and right eigenvectors we would obtain from the SVD (except for possibly flipped signs, which are easy to fix), along with the squared singular values ($e_i = d_i^2$). Note that $\mathbf{T}\mathbf{T}'$ and $\mathbf{T}'\mathbf{T}$ are symmetric, and that the gradients of eigenvectors of symmetric real eigensystems have a simple form (Magnus, 1985, Eq. 7). Furthermore, $\mathbf{T}\mathbf{T}'$ and $\mathbf{T}'\mathbf{T}$ are differentiable wrt. $\mathbf{x}$ and $\mathbf{y}$, enabling a sufficiently efficient implementation in a graph-based, auto-differentiating math compiler[5].

The following section provides a detailed description of the implementation of the CCA layer.

6.7.1 Forward Pass of CCA Projection Layer

For easier reproducibility, we provide a detailed description of the *forward pass* of the proposed CCA layer in Algorithm 1. To train the model, we need to propagate the gradient through the CCA layer (backward pass). We rely on auto-differentiation tools (in particular, *Theano*) implementing the gradient for each individual computation step in the forward pass, and connecting them using the

[4] Note that this is not relevant for the DCCA model introduced by Andrew et al. (2013) because it only derives the CCA projections after optimizing the TNO.

[5] The code of our implementation of the CCA layer is available at `https://github.com/CPJKU/cca_layer`

chain rule.

The layer itself takes the latent feature representations (a batch of m paired vectors $\mathbf{X} \in \mathbb{R}^{d_x \times m}$ and $\mathbf{Y} \in \mathbb{R}^{d_y \times m}$) of the two network pathways f and g as input and projects them with CCA's analytical projection matrices. At train time, the layer uses the optimal projections computed from the current batch. When applying the layer at test time it uses the statistics and projections remembered from last training batch (which can of course be recomputed on a larger training batch to get more stable estimate).

As not all of the computation steps are obvious, we provide further details for the crucial ones. In line 12 and 13, we compute the Cholesky factorization instead of the matrix square root, as the latter has no gradients implemented in *Theano*. As a consequence, we need to transpose $\mathbf{C}_{yy}^{-1}$ when computing $\mathbf{T}$ in line 14 (Hardoon et al., 2004). In line 15 and 16, we compute two eigen decompositions instead of one singular value decomposition (which also has no gradients implemented in *Theano*). In line 19, we flip the signs of first projection matrix to match the second to only have positive correlations. This property is required for retrieval with cosine distance. Finally, in line 24 and 25, the two views get projected using $\mathbf{A}^*$ and $\mathbf{B}^*$. At test time we apply the projections computed and stored during training (line 17).

6.7.2 Investigations on Correlation Structure

As an additional experiment we investigate the correlation structure of the learned representations for all three paradigms. For that purpose we compute the top-most hidden representation $\mathbf{x}$ and $\mathbf{y}$ of the audio-sheet-music-pairs and estimate the canonical correlation coefficients d_i of the respective embedding spaces. For the present example this yields 32 coefficients which is the dimensionality k of our retrieval embedding space. Figure 6.6 compares the correlation coefficients where 1.0 is the maximum value reachable. The most prominent observation in Figure 6.6 is the high correlation coefficients of the representation learned with *DCCA*. This structure is expected as the TNO focuses solely on correlation maximization. However, when recalling the results of Table 6.4 we see that this does not necessarily lead to the best retrieval performance. The freely learned embedding *Learned-*$\mathcal{L}_{rank}$ shows overall the lowest correlation but achieves comparable results to DCCA on this dataset. In terms of overall correlation, CCAL-$\mathcal{L}_{rank}$ is situated in-between the two other approaches. We have seen in all our experiments that combining both concepts in a unified retrieval paradigm yields best retrieval performance over different application domains as well as data regimes. We take this as evidence that componentwise-correlated projections support cosine distance based embedding space learning.

Algorithm 1 Forward Pass of CCA Projection Layer.

Input of layer: $\mathbf{X} \in \mathbb{R}^{d_x \times m}$ and $\mathbf{Y} \in \mathbb{R}^{d_y \times m}$ ▷ hidden representation of current batch
Returns: $\mathbf{X}^*$ and $\mathbf{Y}^*$ ▷ CCA projected hidden representation
Parameters of layer: μ_x, μ_y and $\mathbf{A}^*$, $\mathbf{B}^*$ ▷ means and CCA projection matrices

if train_time **then** ▷ **update statistics and CCA projections during training**

  $\mu_x \leftarrow \frac{1}{m}\sum_i \mathbf{X}_i$ ▷ update μ_x and μ_y with means of batch
  $\mu_y \leftarrow \frac{1}{m}\sum_i \mathbf{Y}_i$

  $\overline{\mathbf{X}} = \mathbf{X} - \mu_x$ ▷ mean center data
  $\overline{\mathbf{Y}} = \mathbf{Y} - \mu_y$

  $\hat{\Sigma}_{xx} = \frac{1}{m-1}\overline{\mathbf{X}}'\overline{\mathbf{X}} + r\mathbf{I}$ ▷ estimate covariances of batch
  $\hat{\Sigma}_{yy} = \frac{1}{m-1}\overline{\mathbf{Y}}'\overline{\mathbf{Y}} + r\mathbf{I}$
  $\hat{\Sigma}_{xy} = \frac{1}{m-1}\overline{\mathbf{X}}'\overline{\mathbf{Y}}$

  $\mathbf{C}_{xx}^{-1} = \text{cholesky}(\hat{\Sigma}_{xx})^{-1}$ ▷ compute inverses of Cholesky factorizations
  $\mathbf{C}_{yy}^{-1} = \text{cholesky}(\hat{\Sigma}_{yy})^{-1}$

  $\mathbf{T} = \mathbf{C}_{xx}^{-1}\hat{\Sigma}_{xy}(\mathbf{C}_{yy}^{-1})'$ ▷ compute matrix $\mathbf{T}$

  $\mathbf{e}, \mathbf{U} = \text{eigen}(\mathbf{T}\mathbf{T}')$ ▷ compute eigenvectors of $\mathbf{T}\mathbf{T}'$ and $\mathbf{T}'\mathbf{T}$
  $\mathbf{e}, \mathbf{V} = \text{eigen}(\mathbf{T}'\mathbf{T})$

  $\mathbf{A}^* \leftarrow \mathbf{C}_{xx}^{-1}\mathbf{U}$ ▷ compute and update CCA projection matrices
  $\mathbf{B}^* \leftarrow \mathbf{C}_{yy}^{-1}\mathbf{V}$

  $\mathbf{A}^* \leftarrow \mathbf{A}^* \cdot \text{sgn}(\text{diag}(\mathbf{A}^{*\prime}\hat{\Sigma}_{xy}\mathbf{B}^*))$ ▷ flip signs of projection matrices

else ▷ **at test time use statistics estimated during training**

  $\overline{\mathbf{X}} = \mathbf{X} - \mu_x$ ▷ mean center test data
  $\overline{\mathbf{Y}} = \mathbf{Y} - \mu_y$

end if

$\mathbf{X}^* = \overline{\mathbf{X}}\mathbf{A}^*$ ▷ project latent representations with CCA projections
$\mathbf{Y}^* = \overline{\mathbf{Y}}\mathbf{B}^*$

return $\mathbf{X}^*$ $\mathbf{Y}^*$

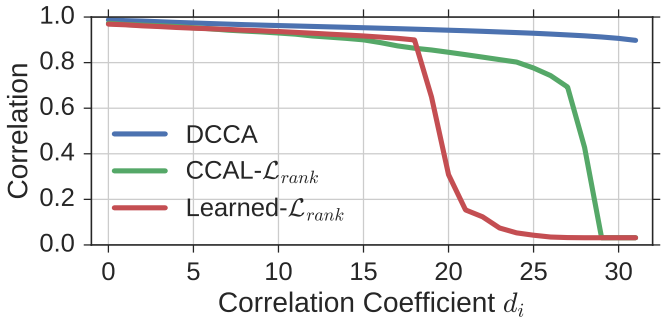


Figure 6.6: Comparison of the 32 correlation coefficients d_i (the dimensionality of the retrieval space is 32) of the topmost hidden representations $\mathbf{x}$ and $\mathbf{y}$ of the audio-to-sheet-music dataset and the respective optimization paradigm. The maximum correlation possible is 1.0 for each coefficient

6.7.3 Architecture and Optimization

In the following we provide additional details for our experiments carried out in Section 6.5.

6.7.3.1 Image-Text Retrieval

We start training with an initial learning rate of either 0.001 (all models on IAPR TC-12 and Flickr30k Learned-$\mathcal{L}_{rank}$) or 0.002 (Flickr30k DCCA and CCAL-$\mathcal{L}_{rank}$)[6]. In addition, we apply 0.0001 $L2$ weight decay and set the batch size to 1000 for all models. The parameter α of the ranking loss in Equation (6.8) is set to 0.5. After no improvement on the validation set for 50 epochs we divide the learning rate by 10 and reduce the patience to 10. This learning rate reduction is repeated three times.

6.7.3.2 Audio-Sheet-Music Retrieval

Table 6.7 provides details on our audio-sheet-music retrieval architecture.

As in the experiments on images and text we optimize our networks using *adam* with an initial learning rate of 0.001 and batch size 1000. The refinement strategy is the same but no weight decay is applied and the margin parameter α of the ranking loss is set to 0.7.

[6]The initial learning rate and parameter α are determined by grid search on the evaluation measure MRR on the validation set.

Table 6.7: Architecture of audio-sheet-music model. BN: Batch Normalization, ELU: Exponential Linear Unit, MP: Max Pooling, Conv(3, pad-1)-16: 3×3 convolution, 16 feature maps and padding 1.

Sheet-Image 40×100	Spectrogram 136×100
2×Conv(3, pad-1)-16	2× Conv(3, pad-1)-16
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-32	2× Conv(3, pad-1)-32
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-64	2× Conv(3, pad-1)-64
BN-ELU + MP(2)	BN-ELU + MP(2)
2×Conv(3, pad-1)-64	2× Conv(3, pad-1)-64
BN-ELU + MP(2)	BN-ELU + MP(2)
Conv(1, pad-0)-32-BN	Conv(1, pad-0)-32-BN
GlobalAveragePooling	GlobalAveragePooling
Respective Optimization Target	

6.7.3.3 Zero-Shot Retrieval

Table 6.8 and 6.9 provide details on the architectures used for our zero-shot retrieval experiments carried out in Section 6.5.4. The general architectures follow Reed et al. (2016) but are optimized with a pairwise ranking loss in combination with our proposed CCA layer. The dimensionality of the retrieval space is fixed to 64 and both models are again optimized with *adam* and a batch size of 1000. The learning rate is set to 0.0007 for the CNN and 0.01 for the CRNN and. The margin parameter α of the ranking loss is set to 0.2. In addition we apply a weight decay of 0.0001 on all trainable parameters of the network for regularization.

Table 6.8: Architecture of Zero-Shot Retrieval CNN. VS: Vocabulary Size, BN: Batch Normalization, ELU: Exponential Linear Unit, MP: Max Pooling, Conv(3, pad-1)-16: 3×3 convolution, 16 feature maps and padding 1.

ImagenNet Feature 1024	Text $VS \times 30 \times 1$
FC(1024)-BN-ELU	1×Conv(3, pad-*same*)-256
FC(1024)-BN-ELU	BN-ELU + MP$(3, 1)$
FC(64)	2×Conv(3, pad-*valid*)-256
	FC(1024)-BN-ELU
	FC(64)
	Respective Optimization Target

Table 6.9: Architecture of Zero-Shot Retrieval CRNN. VS: Vocabulary Size, BN: Batch Normalization, ELU: Exponential Linear Unit, MP: Max Pooling, Conv(3, pad-1)-16: 3×3 convolution, 16 feature maps and padding 1. GRU-RNN: Gated Recurrent Unit (Chung et al., 2014)

ImagenNet Feature 1024	Text $VS \times 30 \times 1$
FC(1024)-BN-ELU	1×Conv(3, pad-*same*)-256
FC(1024)-BN-ELU	BN-ELU + MP$(3, 1)$
FC(64)	2×Conv(3, pad-*valid*)-256
	GRU-RNN(512)
	TemporalAveragePooling
	FC(64)
Respective Optimization Target	

7 Deep Linear Discriminant Analysis

In this chapter I introduce Deep Linear Discriminant Analysis (*DeepLDA*), which learns linearly separable latent representations in an end-to-end fashion. Classic LDA extracts features which preserve class separability, and is used for dimensionality reduction for many classification problems. The central idea is to put LDA on top of a deep neural network. This can be seen as a non-linear extension of classic LDA. Instead of maximizing the likelihood of target labels for individual samples, we propose an objective function that pushes the network to produce feature distributions which: (a) have low variance within the same class and (b) high variance between different classes. Our objective is derived from the general LDA eigenvalue problem and still allows to train with stochastic gradient descent and back-propagation. For evaluation we test our approach on three different benchmark datasets (MNIST, CIFAR-10 and STL-10). DeepLDA produces competitive results on MNIST and CIFAR-10 and outperforms a network trained with categorical cross entropy (having the same architecture) on a supervised setting of STL-10.

This chapter is based on the following publication:

- M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, USA, 2016d.

Personal Contributions The approach presented in this chapter is again inspired by the work of Andrew et al. (2013). I am responsible for developing the method and carried out the experiments.

7.1 Introduction

Linear Discriminant Analysis (LDA) is a method from multivariate statistics which seeks to find a linear projection of high-dimensional observations into a lower-dimensional space (Fisher, 1936). When its preconditions are fulfilled, LDA allows to define optimal linear decision boundaries in the resulting latent space. The aim of this chapter is to exploit the beneficial properties of classic LDA (low intra class variability, hight inter-class variability, optimal decision boundaries) by reformulating its objective to learn linearly separable representations based on a deep neural network (DNN).

Recently, methods related to LDA achieved great success in combination with deep neural networks. Andrew et al. published a deep version of Canonical Correlation Analysis (DCCA) (Andrew et al., 2013). In their evaluations, DCCA is used to produce correlated representations of multimodal input data of simultaneously recorded acoustic and articulatory speech data. Clevert et al. (2015) propose Rectified Factor Networks (RFNs) which are a neural network interpretation of classic factor analysis. RFNs are used for unsupervised pre-training and help to improve classification performance on four different benchmark datasets. A similar method called PCANet – as well as an LDA based variation – was proposed by Chan et al. (2015). PCANet can be seen as a simple unsupervised convolutional deep learning approach. The method proceeds with cascaded Principal Component Analysis (PCA), binary hashing and block histogram computations. However, one crucial bottleneck of their approach is its limitation to very shallow architectures (two stages) (Chan et al., 2015).

Stuhlsatz et. al. already picked up the idea of combining LDA with a neural networks and proposed a generalized version of LDA (Stuhlsatz et al., 2012). Their approach starts with pre-training a stack of restricted Boltzmann machines. In a second step, the pre-trained model is fine-tuned with respect to a linear discriminant criterion. LDA has the disadvantage that it overemphasises large distances at the cost of confusing neighbouring classes. In (Stuhlsatz et al., 2012) this problem is tackled by a heuristic weighting scheme for computing the within-class scatter matrix required for LDA optimization.

Main Idea: The approaches mentioned so far all have in common that they are based on well established methods from multivariate statistics. Inspired by their work, we propose an end-to-end DNN version of LDA - namely *Deep Linear Discriminant Analysis* (*DeepLDA*).

Deep learning has become the state of the art in automatic feature learning and replaced existing approaches based on hand engineered features in many fields such as object recognition (Krizhevsky et al., 2012). DeepLDA is motivated by the fact that when the preconditions of LDA are met, it is capable of finding linear combinations of the input features which allow for optimal linear decision boundaries. In general, LDA takes features as input. The intuition of our method is to use LDA as an objective on top of a powerful feature learning algorithm. Instead of maximizing the likelihood of target labels for individual samples, we propose an LDA eigenvalue-based objective function that pushes the network to produce discriminative feature distributions. The parameters are optimized by back-propagating the error of an LDA-based objective through the entire network. We tackle the feature learning problem by focusing on directions in the latent space with smallest discriminative power. This replaces the weighting scheme of (Stuhlsatz et al.,

2012) and allows to operate on the original formulation of LDA. We expect that DeepLDA will produce linearly separable hidden representations with similar discriminative power in all directions of the latent space. Such representations should also be related with a high classification potential of the respective networks. The experimental classification results reported below will confirm this positive effect on classification accuracy, and two additional experiments (Section 7.5) will give us some first qualitative confirmation that the learned representations show the expected properties.

The reminder of this chapter is structured as follows. In Section 7.2 we provide a general formulation of a DNN. Based on this formulation we introduce DeepLDA, a non-linear extension to classic LDA in Section 7.3. In Section 7.4 we experimentally evaluate our approach on three benchmark datasets. Section 7.5 provides a deeper insight into the structure of DeepLDA's internal represenations. In Section 7.6 we conclude the chapter.

7.2 Deep Neural Networks

As the proposed model is built on top of a DNN we briefly describe the training paradigm of a network used for classification problems such as object recognition.

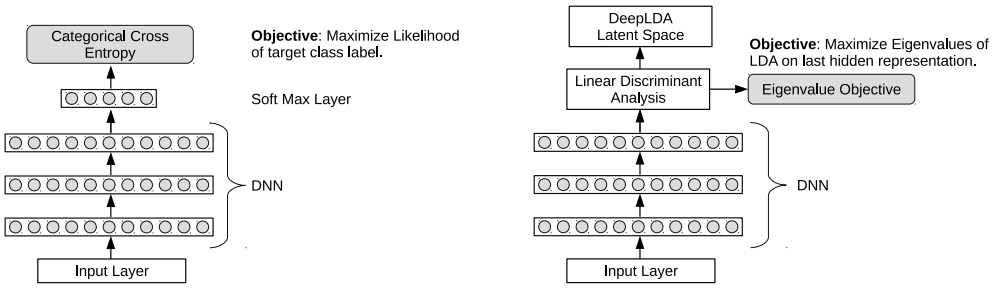
A neural network with P hidden layers is represented as a non-linear function $f(\Theta)$ with model parameters $\Theta = \{\Theta_1, ..., \Theta_P\}$. In the supervised setting we are additionally given a set of N train samples $\mathbf{x}_1, ...\mathbf{x}_N$ along with corresponding classification targets $t_1, ...t_N \in \{1, ..., C\}$. We further assume that the network output $\mathbf{p}_i = (p_{i,1}, ..., p_{i,C}) = f(\mathbf{x}_i, \Theta)$ is normalized by the softmax-function to obtain class (pseudo-)probabilities. The network is then optimized using Stochastic Gradient Descent (SGD) with the goal of finding an optimal model parametrization Θ with respect to a certain loss function $l_i(\Theta) = l(f(\mathbf{x}_i, \Theta), t_i)$.

$$\Theta = \arg\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} l_i(\Theta) \tag{7.1}$$

For multi-class classification problems, Categorical-Cross-Entropy (CCE) is a commonly used optimization target and formulated for observation $\mathbf{x}_i$ and target label t_i as follows

$$l_i(\Theta) = -\sum_{j=1}^{C} y_{i,j} log(p_{i,j}) \tag{7.2}$$

where $y_{i,j}$ is 1 if observation $\mathbf{x}_i$ belongs to class t_i ($j = t_i$) and 0 otherwise. In particular, the CCE tries to maximize the likelihood of the target class t_i for each of the individual training examples $\mathbf{x}_i$ under the model with parameters Θ. Figure 7.1a shows a sketch of this general network architecture.



(a) The output of the network gets normalized by a soft max layer to form valid probabilities. The CCE objective maximizes the likelihood of the target class under the model.

(b) On the topmost hidden layer we compute an LDA which produces corresponding eigenvalues. The optimization target is to maximize those eigenvalues.

Figure 7.1: Schematic sketch of a DNN and DeepLDA. For both architectures the input data is first propagated through the layers of the DNN. However, the final layer and the optimization target are different.

We would like to emphasize that objectives such as CCE do not impose any direct constraints – such as linear separability – on the latent space representation.

7.3 Deep Linear Discriminant Analysis (DeepLDA)

In this section we first provide a general introduction to LDA. Based on this introduction we propose DeepLDA, which optimizes an LDA-based optimization target in an end-to-end DNN fashion. Finally we describe how DeepLDA is used to predict class probabilities of unseen test samples.

7.3.1 Linear Discriminant Analysis

Let $\mathbf{x}_1, ..., \mathbf{x}_N = \mathbf{X} \in \mathbb{R}^{N \times d}$ denote a set of N samples belonging to C different classes $c \in \{1, ..., C\}$. The input representation $\mathbf{X}$ can either be hand engineered features, or hidden space representations $\mathbf{H}$ produced by a DNN (Andrew et al., 2013). LDA seeks to find a linear projection $\mathbf{A} \in \mathbb{R}^{l \times d}$ into a lower l-dimensional subspace L where $l = C - 1$. The resulting linear combinations of features $\mathbf{x}_i \mathbf{A}^T$ are maximally separated in this space (Fisher, 1936). The LDA objective to find projection matrix $\mathbf{A}$ is formulated as:

$$\arg\max_{\mathbf{A}} \frac{|\mathbf{A} S_b \mathbf{A}^T|}{|\mathbf{A} S_w \mathbf{A}^T|} \tag{7.3}$$

where $\mathbf{S}_b$ is the between scatter matrix and defined via the total scatter matrix $\mathbf{S}_t$ and within scatter matrix $\mathbf{S}_w$ as $\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w$. $\mathbf{S}_w$ is defined as the mean of the C individual class covariance matrices $\mathbf{S}_c$ (Equation (7.4) and (7.5)). $\bar{\mathbf{X}}_c = \mathbf{X}_c - \mathbf{m}_c$ are the mean-centered observations of class c with per-class mean vector $\mathbf{m}_c$ ($\bar{\mathbf{X}}$ is defined analogously for the entire population $\mathbf{X}$). The total scatter matrix $\mathbf{S}_t$ is the covariance matrix over the entire population of observations $\mathbf{X}$.

$$\mathbf{S}_c = \frac{1}{N_c - 1} \bar{\mathbf{X}}_c^T \bar{\mathbf{X}}_c \tag{7.4}$$

$$\mathbf{S}_w = \frac{1}{C} \sum_c \mathbf{S}_c \tag{7.5}$$

$$\mathbf{S}_t = \frac{1}{N - 1} \bar{\mathbf{X}}^T \bar{\mathbf{X}} \tag{7.6}$$

The linear combinations that maximize the objective in Equation (7.3) maximize the ratio of between- and within-class scatter also reffered to as separation. This means in particular that a set of projected observations of the same class show low variance, whereas the projections of observations of different classes have high variance in the resulting space L. To find the optimum solution for Equation (7.3) one has to solve the general eigenvalue problem $\mathbf{S}_b\mathbf{e} = \mathbf{v}\mathbf{S}_w\mathbf{e}$. The projection matrix $\mathbf{A}$ is the set of eigenvectors $\mathbf{e}$ associated with this problem. In the following sections we will cast LDA as an objective function for DNN.

7.3.2 DeepLDA Model Configuration

Figure 7.1b shows a schematic sketch of DeepLDA. Instead of sample-wise optimization of the CCE loss on the predicted class probabilities (see Section 7.2) we put an *LDA-layer* on top of the DNN. This means in particular that we do not penalize the misclassification of individual samples. Instead we try to produce features that show a low intra-class and high inter-class variability. We address this maximization problem by a modified version of the general LDA eigenvalue problem proposed in the following section. In contrast to CCE, DeepLDA optimization operates on the properties of the distribution parameters of the hidden representation produced by the neural net. As eigenvalue optimization is tied to its corresponding eigenvectors (a linear projection matrix), DeepLDA can be also seen as a special case of a dense layer.

7.3.3 Modified DeepLDA Optimization Target

Based on Section 7.3.1 we reformulate the LDA objective to be suitable for a combination with deep learning. As already discussed by Stuhlsatz et al. (2012) and Lu et al. (2005) the estimation of $\mathbf{S}_w$ overemphasises high eigenvalues whereas small

eigenvalues are estimated as too low. To weaken this effect, Friedman (1989) proposed to regularize the within scatter matrix by adding a multiple of the identity matrix $\mathbf{S}_w + \lambda\mathbf{I}$. Adding the identity matrix has the second advantage of stabilizing small eigenvalues. The resulting eigenvalue problem is then formulated as

$$\mathbf{S}_b\mathbf{e}_i = v_i(\mathbf{S}_w + \lambda\mathbf{I})\mathbf{e}_i \tag{7.7}$$

where $\mathbf{e} = \mathbf{e}_1, ..., \mathbf{e}_{C-1}$ are the resulting eigenvectors and $\mathbf{v} = v_1, ...v_{C-1}$ the corresponding eigenvalues. Once the problem is solved, each eigenvalue v_i quantifies the amount of discriminative variance (separation) in direction of the corresponding eigenvector $\mathbf{e}_i$. If one would like to combine this objective with a DNN the optimization target would be the maximization of the individual eigenvalues. In particular, we expect that maximizing the individual eigenvalues – which reflect the separation in the respective eigenvector directions – leads to a maximization of the discriminative power of the neural net. In our initial experiments we started to formulate the objective as:

$$\arg\max_{\Theta} \frac{1}{C-1} \sum_{i=1}^{C-1} v_i \tag{7.8}$$

One problem we discovered with the objective in Equation (7.8) is that the net favours trivial solutions e. g. maximize only the largest eigenvalue as this produces the highest reward. In terms of classification this means that it maximizes the distance of classes that are already separated at the expense of – potentially non-separated – neighbouring classes. This was already discussed by (Stuhlsatz et al., 2012) and tackled by a weighted computation of the between scatter matrix $\mathbf{S}_b$.

We propose a different solution to this problem and address it by focusing our optimization on the smallest of all $C-1$ available eigenvalues. In particular we consider only the k eigenvalues that do not exceed a certain threshold for variance maximization:

$$\arg\max_{\Theta} \frac{1}{k} \sum_{i=1}^{k} v_i \;\text{ with }\; \{v_1, ..., v_k\} = \{v_j | v_j < \min\{v_1, ..., v_{C-1}\} + \epsilon\} \tag{7.9}$$

The intuition behind this formulation is to learn a net parametrization that pushes as much discriminative variance as possible into all of the $C-1$ available feature dimensions.

We would like to underline that this formulation allows to train DeepLDA networks with back-propagation in end-to-end fashion (see Appendix for a derivative of the loss functions's gradient). Our models are optimized with the Nesterov momentum version of mini-batch SGD. Related methods already showed that mini-batch learning on distribution parameters (in this case covariance matrices) is feasible if the batch-size is sufficiently large to be representative for the entire population (Wang et al., 2015a,b).

7.3.4 Classification by DeepLDA

This section describes how the most likely class label is assigned to an unseen test sample $\mathbf{x}_t$ once the network is trained and parametrized. In a first step we compute the topmost hidden representation $\mathbf{H}$ on the entire training set $\mathbf{X}$. On this hidden representation we compute the LDA as described in Section 7.3.1 and 7.3.3 producing the corresponding eigenvectors $\mathbf{e} = \{\mathbf{e}_i\}_{i=1}^{C-1}$ which form the LDA projection matrix $\mathbf{A}$. We would like to emphasize that since the parameters of the network are fixed at this stage we make use of the entire training set to provide a stable estimate of the LDA projection. Based on $\mathbf{A}$ and the per-class mean hidden representations $\bar{\mathbf{H}}_c = (\bar{\mathbf{h}}_1^T, ..., \bar{\mathbf{h}}_C^T)$ the distances of sample $\mathbf{h}_t$ to the linear decision hyperplanes (Friedman et al., 2001) are defined as

$$\mathbf{d} = \mathbf{h}_t^T \mathbf{T}^T - \frac{1}{2} diag\left(\bar{\mathbf{H}}_c \mathbf{T}^T\right) \text{ with } \mathbf{T} = \bar{\mathbf{H}}_c \mathbf{A} \mathbf{A}^T \tag{7.10}$$

where $\mathbf{T}$ are the decision hyperplane normal vectors. The subtracted term is the bias of the decision functions placing the decision boundaries in between the means of the respective class hidden representations (no class priors included). The vector of class probabilities for test sample $\mathbf{x}_t$ is then computed by applying the logistic function $\mathbf{p}'_c = 1/(1 + \mathrm{e}^{-\mathbf{d}})$ and further normalized by $\mathbf{p}_c = \mathbf{p}'_c / \sum p'_i$ to sum to one. Finally we assign class i with highest probability as $\arg\max_i p_i$ to the unseen test sample $\mathbf{x}_t$.

7.4 Experiments

In this section we present an experimental evaluation of DeepLDA on three benchmark data sets – namely MNIST, CIFAR-10 and STL-10 (see Figure 7.2 for some sample images). We compare the results of DeepLDA with the CCE based optimization target as well as the present state of the art of the respective datasets. In addition, we provide details on the network architectures, hyper parameters and respective training/optimization approaches used in our experiments.

7.4.1 Experimental Setup

The general structure of the networks is similar for all of the three datasets and identical for CIFAR-10 and STL-10. The architecture follows the VGG model with sequences of 3×3 convolutions (Simonyan and Zisserman, 2015). Instead of a dense classification layer we use global average pooling on the feature maps of the last convolution layer (Lin et al., 2014). We picked this architecture as it leads to well-posed problems for covariance estimation: many samples vs. low feature space dimension. We further apply batch normalization (Ioffe and Szegedy, 2015) after

(a) (b) (c) (d) (e) (f)

Figure 7.2: Example images of evaluation data sets (a)(b) MNIST, (c)(d) CIFAR-10, (e)(f) STL-10. The relative size differences between images from the three data sets are kept in this visualization.

each convolutional layer which (1) helped to increase convergence speed and (2) improved the performance of all our models. Batch normalization has a positive effect on both CCE as well as DeepLDA-based optimization. In Table 7.1 we outline the structure of our models in detail. All networks are trained using SGD with Nesterov momentum. The initial learning rate is set to 0.1 and the momentum is fixed at 0.9 for all our models. The learning rate is then halved every 25 epochs for CIFAR-10 and STL-10 and every 10 epochs for MNIST. For further regularization we add weight decay with a weighting of 0.0001 on all trainable parameters of the models. The between-class covariance matrix regularization weight λ (see Section 7.3.3) is set to 0.001 and the ϵ-offset for DeepLDA to 1.

One hyper-parameter that varies between the datasets is the batch size used for training DeepLDA. Although a large batch size is desired to get stable covariance estimates it is limited by the amount of memory available on the GPU. The mini-batches for DeepLDA were for MNIST: 1000, for CIFAR-10: 1000 and for STL-10: 200. For CCE training, a batch size of 128 is used for all datasets. The models are trained on an NVIDIA Tesla K40 with 12GB of GPU memory.

7.4.2 Experimental Results

We describe the benchmark datasets as well as the pre-processing and data augmentation used for training. We present our results and relate them to the present state of the art for the respective dataset. As DeepLDA is supposed to produce a linearly separable feature space, we also report the results of a linear Support Vector Machine trained on the latent space of DeepLDA (tagged with *LinSVM*). The results of our network architecture trained with CCE are marked as *OurNetCCE*. To provide a complete picture of our experimental evaluation we also show classification results of an LDA on the topmost hidden representation of the networks trained with CCE (tagged with *OurNetCCE(LDA)*).

Table 7.1: Model Specifications. BN: Batch Normalization, ReLu: Rectified Linear Activation Function, CCE: Categorical Cross Entropy. The mini-batch sizes of DeepLDA are: MNIST(1000), CIFAR-10(1000), STL-10(200). For CCE training a constant batch size of 128 is used.

CIFAR-10 and STL-10	MNIST
Input $3 \times 32 \times 32$ (96×96)	Input $1 \times 28 \times 28$
3×3 Conv(pad-1)-64-BN-ReLu 3×3 Conv(pad-1)-64-BN-ReLu 2×2 Max-Pooling + Drop-Out(0.25)	
3×3 Conv(pad-1)-128-BN-ReLu 3×3 Conv(pad-1)-128-BN-ReLu 2×2 Max-Pooling + Drop-Out(0.25)	3×3 Conv(pad-1)-96-BN-ReLu 3×3 Conv(pad-1)-96-BN-ReLu 2×2 Max-Pooling + Drop-Out(0.25)
3×3 Conv(pad-1)-256-BN-ReLu 3×3 Conv(pad-1)-256-BN-ReLu 3×3 Conv(pad-1)-256-BN-ReLu 3×3 Conv(pad-1)-256-BN-ReLu 2×2 Max-Pooling + Drop-Out(0.25)	
3×3 Conv(pad-0)-1024-BN-ReLu Drop-Out(0.5)	3×3 Conv(pad-0)-256-BN-ReLu Drop-Out(0.5)
1×1 Conv(pad-0)-1024-BN-ReLu Drop-Out(0.5)	1×1 Conv(pad-0)-256-BN-ReLu Drop-Out(0.5)
1×1 Conv(pad-0)-10-BN-ReLu 2×2 (10×10) Global-Average-Pooling	1×1 Conv(pad-0)-10-BN-ReLu 5×5 Global-Average-Pooling
Soft-Max with CCE or LDA-Layer	

7.4.2.1 MNIST

The MNIST dataset consists of 28×28 gray scale images of handwritten digits ranging from 0 to 9. The dataset is structured into 50000 train samples, 10000 validation samples and 10000 test samples. For training we did not apply any pre-processing nor data augmentation. We present results for two different scenarios. In scenario *MNIST-50k* we train on the 50000 train samples and use the validation set to pick the parametrization which produces the best results on the validation set. In scenario *MNIST-60k* we train the model for the same number of epochs as in *MNIST-50k* but also use the validation set for training. Finally we report the accuracy of the model on the test set after the last training epoch. This approach was also applied in (Lin et al., 2014) which produce state of the art results on the dataset.

Table 7.2 summarizes all results on the MNIST dataset. DeepLDA produces

Table 7.2: Comparison of test errors on MNIST

Method	Test Error
NIN + Dropout (Lin et al. (2014))	0.47%
Maxout (Goodfellow et al. (2013))	0.45%
DeepCNet(5,60) (Graham (2014))	0.31% (train set translation)
OurNetCCE(LDA)-50k	0.39%
OurNetCCE-50k	0.37%
OurNetCCE-60k	0.34%
DeepLDA-60k	0.32%
OurNetCCE(LDA)-60k	0.30%
DeepLDA-50k	**0.29**%
DeepLDA-50k(LinSVM)	**0.29**%

competitive results – having a test set error of 0.29% – although no data augmentation is used. In the approach described in (Graham, 2014) the train set is extended with translations of up to two pixels. We also observe that a linear SVM trained on the learned representation produces comparable results on the test set. It is also interesting that early stopping with best-model-selection (MNIST-50k) performs better than training on MNIST-60k even though 10000 more training examples are available.

7.4.2.2 CIFAR-10

The CIFAR-10 dataset consists of tiny 32×32 natural RGB images containing samples of 10 different classes. The dataset is structured into 50000 train samples and 10000 test samples. We pre-processed the dataset using global contrast normalization and ZCA whitening as proposed by Goodfellow et al. (2013). During training we only apply random left-right flips on the images – no additional data augmentation is used. In training, we follow the same procedure as described for the MNIST dataset above to make use of the entire 50000 train images.

Table 7.3 summarizes our results and relates them to the present state of the art. Both OurNetCCE and DeepLDA produce state of the art results on the dataset when no data augmentation is used. Although DeepLDA performs slightly worse than CCE it is capable of producing competitive results on CIFAR-10.

Table 7.3: Comparison of test errors on CIFAR-10

Method	Test Error
NIN + Dropout (Lin et al. (2014))	10.41%
Maxout (Graham (2014))	9.38%
NIN + Dropout (Lin et al. (2014))	8.81% (data augmentation)
DeepCNINet(5,300) (Graham (2014))	**6.28**% (data augmentation)
DeepLDA(LinSVM)	7.58%
DeepLDA	7.29%
OurNetCCE(LDA)	7.19%
OurNetCCE	7.10%

7.4.2.3 STL-10

Like CIFAR-10, the STL-10 data set contains natural RGB images of 10 different object categories. However, with 96×96 pixels the size of the images is larger, and the training set is considerably smaller, with only 5000 images. The test set consists of 8000 images. In addition, STL-10 contains 100000 unlabelled images but we do not make use of this additional data at this point as our approach is fully supervised. For that reason we first perform an experiment (*Method-4k*) where we do not follow the evaluation strategy described in (Coates et al., 2011), where models are trained on 1000 labeled and 100000 unlabeled images. Instead, we directly compare CCE and DeepLDA in a fully supervised setting. As with MNIST-50k we train our models on 4000 of the train images and use the rest (1000 images) as a validation set to pick the best performing parametrization. The results on the *Method-4k-Setting* of STL-10 are presented in the top part of Table 7.4. Our model trained with CCE achieves an accuracy of 78.39%. The same architecture trained with DeepLDA improves the test set accuracy by more than 3 percentage points and achieves 81.46%. In our second experiment (*Method-1k*) we follow the evaluation strategy described in (Coates et al., 2011) but without using the unlabelled data. We train our models on the 10 pre-defined folds (each fold contains 1000 train images) and report the average accuracy on the test set. The model optimized with CCE (*OurNetCCE-1k*) achieves 57.44% accuracy on the test set which is in line with the supervised results reported in (Zhao et al., 2015).

Our model trained with DeepLDA achieves 66.97% average test set accuracy. This is a performance gain of 9.53% in contrast to CCE and it shows that the advantage of DeepLDA compared to CCE becomes even more apparent when the amount of labeled data is low. When comparing *DeepLDA-1k* with LDA applied on the features computed by a network trained with CCE (*OurNetCCE(LDA)-1k*,

Table 7.4: Comparison of test set accuracy on a purely supervised setting of STL-10. (*Method-4k*: 4000 train images, *Method-1k*: 1000 train images.)

Method-4k	Test Accuracy-4k
OurNetCCE(LDA)-4k	78.50%
OurNetCCE-4k	78.84%
DeepLDA-4k	81.16%
DeepLDA(LinSVM)-4k	**81.40**%

Method-1k	Test Accuracy-1k
SWWAE (Zhao et al. (2015))	57.45%
SWWAE (Zhao et al. (2015))	74.33% (semi-supervised)
DeepLDA(LinSVM)-1k	55.92%
OurNetCCE-1k	57.44%
OurNetCCE(LDA)-1k	59.48%
DeepLDA-1k	**66.97**%

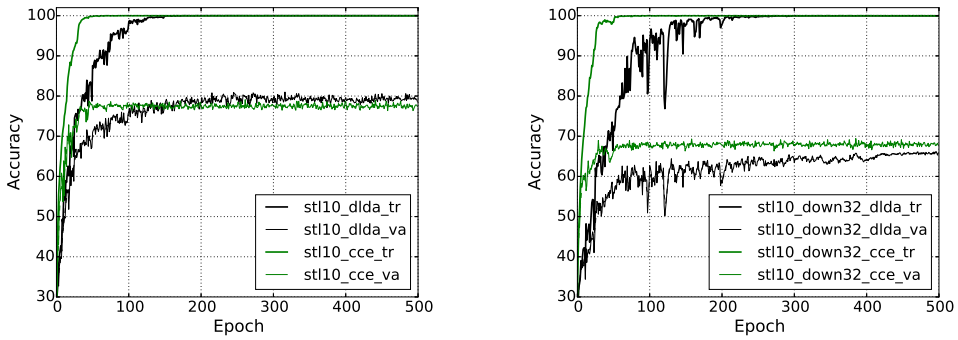
59.48%), we find that the end-to-end trained LDA-features outperform the standard CCE approach. A direct comparison with state of the art results as reported in (Zhao et al., 2015; Swersky et al., 2013; Dosovitskiy et al., 2014) is not possible because these models are trained under semi-supervised conditions using both unlabelled and labelled data. However, the results suggest that a combination of DeepLDA with methods such as proposed by Zhao et al. (2015) is a very promising future direction.

7.5 Investigatons on DeepLDA and Discussions

In this section we provide deeper insights into the representations learned by DeepLDA. We experimentally investigate the eigenvalue structure of representations learned by DeepLDA as well as its relation to the classification potential of the respective networks.

7.5.1 Does Image Size Affect DeepLDA?

DeepLDA shows its best performance on the STL-10 dataset (*Method-4k*) where it outperforms CCE by 3 percentage points. The major difference between STL-10 and CIFAR-10 – apart from the number of train images – is the size of the contained images (see Figure 7.2 to get an impression of the size relations). To get



(a) STL-10 (96×96)

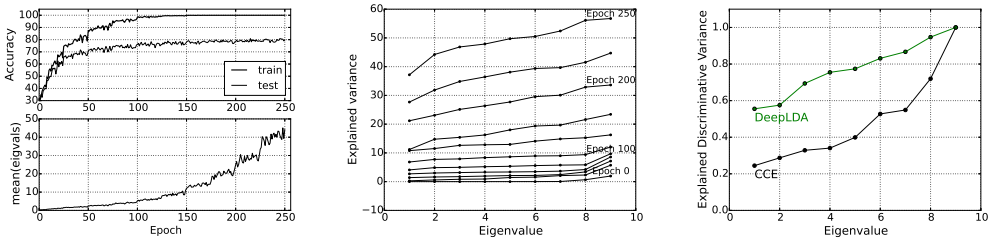
(b) STL-10 (32×32)

Figure 7.3: Comparison of the learning curves of DeepLDA on the original STL-10 dataset (*Method-4k*) with image size 96×96 and its downscaled 32×32 version.

a deeper insight into the influence of this parameter we run the following additional experiment: (1) we create a downscaled version of the STL-10 dataset with the same image dimensions as CIFAR-10 (32×32). (2) We repeat the experiment (*Method-4k*) described in Section 7.4.2.3 on the downscaled 32×32 dataset. The results are presented in Figure 7.3, as curves showing the evolution of train and validation accuracy during training. As expected, downscaling reduces the performance of both CCE and DeepLDA. We further observe that DeepLDA performs best when trained on larger images and has a disadvantage on the small images. However, a closer look at the results on CIFAR-10 (CCE: 7.10% error, DeepLDA: 7.29% error, see Table 7.3) suggests that this effect is compensated when the training set size is sufficiently large. As a reminder: CIFAR-10 contains 50000 train images in contrast to STL-10 with only 4000 samples.

7.5.2 Eigenvalue Structure of DeepLDA Representations

DeepLDA optimization does not focus on maximizing the target class likelihood of individual samples. As proposed in Section 7.3 we encourage the net to learn feature representations with discriminative distribution parameters (within and between class scatter). We achieve this by exploiting the eigenvalue structure of the general LDA eigenvalue problem and use it as a deep learning objective. Figure 7.4a shows the evolution of train and test set accuracy of STL-10 along with the mean value of all eigenvalues in the respective training epoch. We observe the expected natural correlation between the magnitude of explained "discriminative" variance (separation) and the classification potential of the resulting representation. In Fig-



(a) Eigenvalues vs. Accuracy (b) Individual Eigenvalues (c) CCE vs. DeepLDA

Figure 7.4: The figure investigates the eigenvalue structure of the general LDA eigenvalue problem during training a DeepLDA network on STL-10 (*Method-4k*). (a) shows the evolution of classification accuracy along with the magnitude of explained discriminative variance (separation) in the latent representation of the network. (b) shows the evolution of individual eigenvalues during training. In (c) we compare the eigenvalue structure of a net trained with CCE and DeepLDA (for better comparability we normalized the maximum eigenvalue to one).

ure 7.4b we show how the individual eigenvalues increase during training. Note that in Epoch 0 almost all eigenvalues (1-7) start at a value of 0. This emphasizes the importance of the design of our objective function (compare Equation (7.9)) which allows to draw discriminability into the lower dimensions of the eigen-space. In Figure 7.4c we additionally compare the eigenvalue structure of the latent representation produced by DeepLDA with CCE based training. Again results show that DeepLDA helps to distribute the discriminative variance more equally over the available dimensions. To give the reader an additional intuition on the learned representations we visualize the latent space of STL-10 in our supplemental materials on the final page of this chapter.

7.6 Conclusion

We have presented DeepLDA, a deep neural network interpretation of linear discriminant analysis. DeepLDA learns linearly separable latent representations in an end-to-end fashion by maximizing the eigenvalues of the general LDA eigenvalue problem. Our modified version of the LDA optimization target pushes the network to distribute discriminative variance in all dimensions of the latent feature space. Experimental results show that representations learned with DeepLDA are discriminative and have a positive effect on classification accuracy. Our DeepLDA models achieve competitive results on MNIST and CIFAR-10 and outperform CCE in a

fully supervised setting of STL-10 by more than 9% test set accuracy. The results and further investigations suggest that DeepLDA performs best, when applied to reasonably-sized images (in the present case 96×96 pixel). Finally, we see DeepLDA as a specific instance of a general fruitful strategy: exploit well-understood machine learning or classification models such as LDA with certain desirable properties, and use deep networks to learn representations that provide optimal conditions for these models.

7.7 Appendix A: Gradient of DeepLDA-Loss

To train with back-propagation we provide the partial derivatives of optimization target $l(\mathbf{H})$ proposed in Equation (7.9) with respect to the topmost hidden representation $\mathbf{H}$ (contains samples as rows and features as columns). As a reminder, the DeepLDA objective focuses on maximizing the k smallest eigenvalues v_i of the generalized LDA eigenvalue problem. In particular, we consider only the k eigenvalues that do not exceed a certain threshold for optimization:

$$l(\mathbf{H}) = \frac{1}{k}\sum_{i=1}^{k} v_i \text{ with } \{v_1, ..., v_k\} = \{v_j | v_j < \min\{v_1, ..., v_{C-1}\} + \epsilon\} \tag{7.11}$$

For convenience, we change the subscripts of the scatter matrices to superscripts in this section (e.g. $\mathbf{S}_t \rightarrow \mathbf{S}^t$). $\mathbf{S}^t_{ij}$ addresses the element in row i and column j in matrix $\mathbf{S}^t$. Starting from the formulation of the generalized LDA eigenvalue problem:

$$\mathbf{S}^b\mathbf{e}_i = v_i\mathbf{S}^w\mathbf{e}_i \tag{7.12}$$

the derivative of eigenvalue v_i with respect to hidden representation $\mathbf{H}$ is defined in (de Leeuw, 2007) as:

$$\frac{\partial v_i}{\partial \mathbf{H}} = \mathbf{e}_i^T \left(\frac{\partial \mathbf{S}^b}{\partial \mathbf{H}} - v_i \frac{\partial \mathbf{S}^w}{\partial \mathbf{H}} \right) \mathbf{e}_i \tag{7.13}$$

Recalling the definitions of the LDA scatter matrices from Section 7.3.1:

$$\mathbf{S}^c = \frac{1}{N_c - 1}\bar{\mathbf{X}}_c^T\bar{\mathbf{X}}_c \qquad \mathbf{S}^w = \frac{1}{C}\sum_c \mathbf{S}^c \tag{7.14}$$

$$\mathbf{S}^t = \frac{1}{N-1}\bar{\mathbf{X}}^T\bar{\mathbf{X}} \qquad \mathbf{S}^b = \mathbf{S}^t - \mathbf{S}^w \tag{7.15}$$

we can write the partial derivative of the total scatter matrix $\mathbf{S}_t$ (Andrew et al., 2013; Stuhlsatz et al., 2012) on hidden representation $\mathbf{H}$ as:

$$\frac{\partial \mathbf{S}^t_{ab}}{\partial H_{ij}} = \begin{cases} \frac{2}{N-1}\left(H_{ij} - \frac{1}{N}\sum_n H_{nj}\right) & \text{if } a = j, b = j \\ \frac{1}{N-1}\left(H_{ib} - \frac{1}{N}\sum_n H_{nb}\right) & \text{if } a = j, b \neq j \\ \frac{1}{N-1}\left(H_{ia} - \frac{1}{N}\sum_n H_{na}\right) & \text{if } a \neq j, b = j \\ 0 & \text{if } a \neq j, b \neq j \end{cases} \tag{7.16}$$

The derivatives for the individual class covariance matrices $\mathbf{S}^c$ are defined analogously to Equation (7.16) for the C classes and we can write the partial derivatives of $\mathbf{S}^w$ and $\mathbf{S}^b$ with respect to the latent representation $\mathbf{H}$ as:

$$\frac{\partial \mathbf{S}^w_{ab}}{\partial H_{ij}} = \frac{1}{C}\sum_c \frac{\partial \mathbf{S}^c_{ab}}{\partial H_{ij}} \quad \text{and} \quad \frac{\partial \mathbf{S}^b_{ab}}{\partial H_{ij}} = \frac{\partial \mathbf{S}^t_{ab}}{\partial H_{ij}} - \frac{\partial \mathbf{S}^w_{ab}}{\partial H_{ij}} \tag{7.17}$$

The partial derivative of the loss function introduced in Section 7.3.3 with respect to hidden state $\mathbf{H}$ is then defined as:

$$\frac{\partial}{\partial \mathbf{H}} \frac{1}{k} \sum_{i=1}^{k} v_i = \frac{1}{k} \sum_{i=1}^{k} \frac{\partial v_i}{\partial \mathbf{H}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{e}_i^T \left(\frac{\partial \mathbf{S}^b}{\partial \mathbf{H}} - v_i \frac{\partial \mathbf{S}^w}{\partial \mathbf{H}} \right) \mathbf{e}_i \tag{7.18}$$

7.8 Appendix B: DeepLDA Latent Representation

Figure 7.5 shows the latent space representations on the STL-10 dataset (Method-4k) as *n-to-n* scatter plots of the latent features on the first 1000 test set samples. We plot the test set samples after projection into the $C-1$ dimensional DeepLDA feature space. The plot suggest that DeepLDA makes use of all available feature dimensions. An interesting observation is that many of the internal representations are orthogonal to each other (which is an implication of LDA). This of course favours linear decision boundaries.

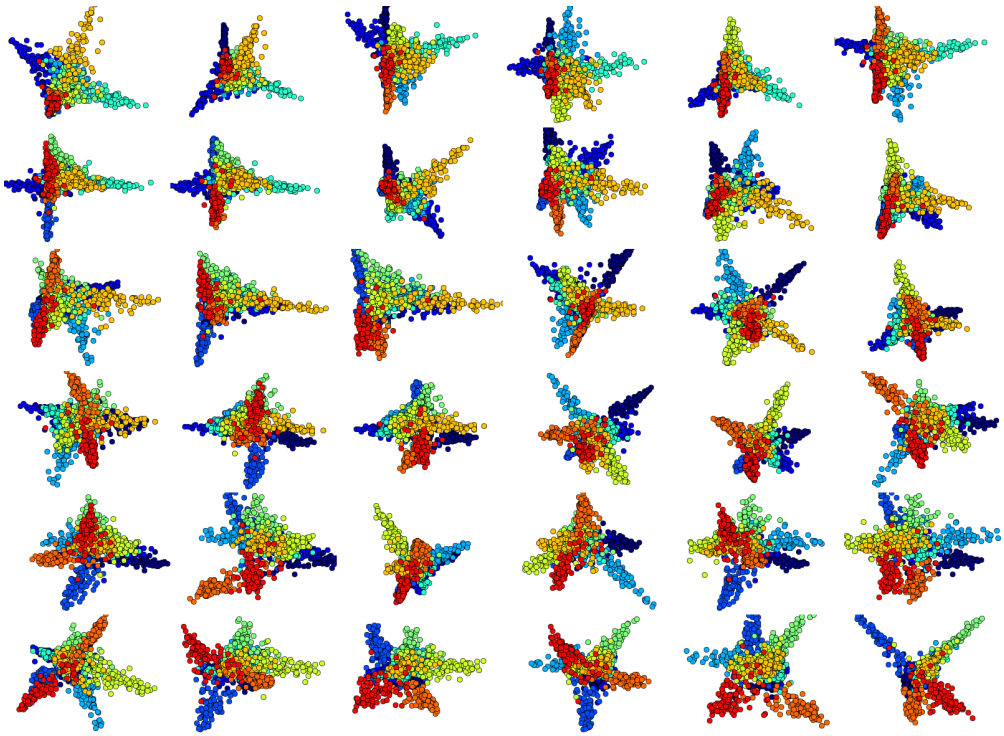


Figure 7.5: STL-10 latent representation produced by DeepLDA (*n-to-n* scatter plots of the latent features of the first 1000 test set samples. e.g.: top left plot: latent feature 1 vs. latent feature 2).

Part III

Conclusions and Future Work

In the final part of my thesis I try to give a complete picture by summarizing the main findings and conclusions across all parts and chapters. I also point at open research problems and directions which are in my opinion the most prominent and promising ones to be tackled as next steps.

Convolutional Networks Networks for Sheet Music Images. Chapters 3, 4, and 5 of this thesis show how to utilize three different machine learning paradigms for learning task-specific representations of audio and sheet music. All of this started with a proof of concept on simple monophonic melodies and supervised function approximation by formulating score following in sheet music images as a localization task. In a next stage this evolved to multimodal joint embedding space learning for cross-modality retrieval of complex piano music. Given the learned joint embedding space we are able to address tasks such as piece identification and offline audio – sheet music alignment. For the last class of machine learning methods, reinforcement learning, we design agents which learn the task of score following in sheet music images purely via interacting with their environment, the score following game.

What all three approaches have in common is their core component, a multimodal convolutional neural network (CNN) processing both sheet music images and audio at the same time. I conclude, given the successful application of multimodal CNNs on these three diverse tasks as strong empirical evidence, that CNNs are the right choice for addressing sheet-music-related problems. This is also supported by advances in related field such as optical music recognition where convolution networks also became a successful choice (Calvo-Zaragoza and Rizo, 2018; van der Wel and Ullrich, 2017) in recent years. As a final remark, I would like to note that the audio – sheet music domain has been a very exciting machine learning playground for me which leaves many challenging open problems for further research. I will discuss the most prominent ones below.

Data is Key. As already discussed, all approaches presented in this thesis have in common that they are built around neural networks that learn their behavior purely from training observations. There is a lot of experimental evidence, especially in the computer vision community, that in order to arrive at general models, a sufficiently large and representative training set is required that contains as little label noise (annotation errors) as possible (Zhang et al., 2017). We have seen in the experiments carried out in Part I that the MSMD dataset proposed in this thesis is an important step towards such a dataset for audio and sheet music. Although MSMD is entirely synthetic (rendered sheet music, synthetic audio) we arrive at models that start to generalize far beyond this synthetic domain for the task of piece identification. Evaluations on this identification task on realistic data

(scans of sheet music and audio recordings of real performances) are feasible as no strong labels are required – it is sufficient to have a piece-level association between score/piece and recording/performance. Strong labels refer here to one-to-one correspondences between audio and sheet music image. However, when we want to evaluate alignment or precise localization tasks we need precise, point-wise alignment ground truth between scanned sheet music and audios recorded from real performance.

Therefore, the next logical step in this line of research is, in my opinion, to collect a comprehensive real world dataset by aligning audio recordings from real performances of both professional and amateur performers, with scanned images of sheet music. In the beginning, this does not necessarily have to be a large scale dataset which would be practical for training robust models. Already a challenging evaluation set, allowing to investigate if the models developed on synthetic data generalize to this realistic domain, will be a valuable resource. Collecting such a dataset is also in direct relation to the following paragraph on *robust audio representations*.

Robust Audio Representation Learning. Recall that all models introduced in Part I have a fixed and limited field of view on both the sheet music and the audio. In particular, these fields of view define the input dimensionality of the respective pathways of the multimodal convolutional networks. While this is less crucial for the sheet music side, it definitely is an issue for the audio, especially when dealing with realistic performance audios. To give a simple example: a human listener can easily recognize the same piece of music played in very different tempi, but when the audio is segmented into spectrogram excerpts with a fixed number of time steps, these contain disparate amounts of musical content, relative to what the model has seen during training. Besides this trivial example, expressive performances may also exhibit rather extreme tempo changes in addition to challenges such as asynchronous onsets, pedal, room acoustics, or dynamics (Widmer, 2017).

Given these facts about performances and our experimental findings on Chopin recordings of professional pianists (Chapter 4), I conclude that learning robust audio representations is one of the main open research problems to be addressed. At the time writing this thesis I can not yet suggest a concrete solution to all of these problems but I expect recurrent neural networks and especially attention mechanisms (Vaswani et al., 2017; Chan et al., 2016; Xu et al., 2015; Bahdanau et al., 2014) to be fruitful research direction at least to tackle the problem of tempo invariance.

Reinforcement Learning as a Perspective for Music Information Retrieval. In Chapter 5, I show how to utilize recent advances in deep reinforcement learning to train agents which are capable of following the currently playing music audio

along in the respective images of sheet music. The desired behavior of an agent is specified simply by designing an appropriate reward function. Given this reward definition, the agent then learns by interacting with its environment. Although this is only a first step taken for one particular application, I would like to emphasize the potential of this very general machine learning paradigm for the music domain. In my opinion the two most promising characteristics are the following. Firstly, we can learn from a delayed and potentially sparse reward signal. This implies that annotations on a much coarser level are sufficient at training time. Sticking to the task of score following, we could for example annotate a set of real performance audios at the bar level, which is perfectly feasible. Reward is then only generated for those time steps where an annotation is present. Secondly, we are much less restricted when designing objective functions for optimizing our networks. To also provide an example for this claim: Instead of training a piano transcription model in a supervised fashion by regressing ground truth targets (the MIDI) we could provide an agent with an instrument (e.g. a piano synthesizer). The task is then to learn, via trial and error, to minimize the difference between a certain target audio at hand and what is proposed (played) by the agent. If this difference between target audio and the agent's proposal approaches zero the agent has learned to reproduce the piece, which implicitly yields the correct transcription. Note that this transcription concept does not require any labels at all (Kelz and Widmer, 2018).

Classical Multivariate Statistics and Deep Learning. The second part of this thesis focuses on the extension and reformulation of methods from classical multivariate statistics to make them compatible with deep neural networks. I already stated, when working on DeepLDA, that I consider exploiting well-understood machine learning or classification models with certain desirable properties, as a general fruitful strategy to come up with new models. In such settings, deep networks serve as representation learners that provide optimal conditions for these models. Related work focusing for example on Factor Analysis (Clevert et al., 2015) or CCA (Andrew et al., 2013) also supports this conclusion and besides that, heavily inspired my work. Especially the latter was the basis for the CCA-Layer proposed in Chapter 6. Our retrieval experiments revealed that CCA-based models consistently outperform the models that learn the embedding projections from scratch. A closer look at the results further showed that the CCA layer acts as a strong regularizer to prevent over-fitting especially in the low data regime. Given the findings of Part II of this thesis, I am convinced that it is worth to further explore the potential of well understood methods such as LDA or CCA in the context of deep neural networks.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255, Atlanta, USA, 2013.

A. Arzt. *Flexible and robust music tracking.* doctoralthesis, Johannes Kepler Universität Linz, Linz, 2016.

A. Arzt, G. Widmer, and S. Dixon. Automatic page turning for musicians via real-time machine listening. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 241–245, Patras, Greece, 2008.

A. Arzt, H. Frostel, T. Gadermaier, M. Gasser, M. Grachten, and G. Widmer. Artificial intelligence in the concertgebouw. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2424–2430, Buenos Aires, Argentina, 2015.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

S. Balke. *Multimedia Processing Techniques for Retrieving, Extracting, and Accessing Musical Content.* doctoralthesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany, 2018.

S. Balke, S. P. Achankunju, and M. Müller. Matching musical themes based on noisy OCR and OMR input. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 703–707, Brisbane, Australia, 2015.

S. Balke, V. Arifi-Müller, L. Lamprecht, and M. Müller. Retrieving audio recordings using musical themes. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 281–285, Shanghai, China, 2016.

J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Austin, USA, 2010.

S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 121–124, Kyoto, Japan, 2012.

S. Böck, F. Korzionwski, J. Schlüter, F. Krebs, and G. Widmer. madmom: A new python audio and music signal processing library. In *Proceedings of the 2016 ACM Multimedia Conference*, pages 1174–1178, Amsterdam, The Netherlands, 2016.

N. Boulanger-lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1159–1166, Edinburgh, Scotland, 2012.

D. Byrd and J. G. Simonsen. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, 44(3):169–195, 2015. doi:10.1080/09298215.2015.1045424.

J. Calvo-Zaragoza and D. Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4):472–477, 2018. doi:10.3390/app8040606.

M. A. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):1015–1028, 2008. doi:10.1109/TASL.2008.925883.

T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24 (12):5017–5032, 2015. doi:10.1109/TIP.2015.2475625.

W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4960–4964, Shanghai, China, 2016.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *In Proceedings of the British Machine Vision Conference*, pages 1–5, Nottingham, UK, 2014.

T. Cheng, M. Mauch, E. Benetos, and S. Dixon. An attack/decay model for piano transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 584–590, New York City, USA, 2016.

J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

D.-A. Clevert, T. Unterthiner, A. Mayr, H. Ramsauer, and S. Hochreiter. Rectified factor networks. In *Advances in neural information processing systems (NIPS)*, pages 1855–1863, Montreal, Canada, 2015.

A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics (AISTATS)*, pages 215–223, Fort Lauderdale, USA, 2011.

A. Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6): 837–846, 2009. doi:10.1109/TPAMI.2009.106.

N. Cook. Performance analysis and chopin's mazurkas. *Musicae Scientae*, 11(2): 183–205, 2007.

R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 193–198, Paris, France, 1984.

J. de Leeuw. Derivatives of generalized eigen systems with applications. 2007.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, USA, 2009.

S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, E. Battenberg, A. van den Oord, et al. Lasagne: First release., 2015.

M. Dorfer and J. Mattes. Recursive water flow: A shape decomposition approach for cell clump splitting. In *13th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 811–815, Prague, Czech Republic, 2016.

M. Dorfer and G. Widmer. Towards deep and discriminative canonical correlation analysis. In *ICML 2016 Workshop on Multi-View Representation Learning*, New York, USA, 2016a.

M. Dorfer and G. Widmer. Towards end-to-end audio-sheet-music retrieval. In *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, Barcelona, Spain, 2016b.

M. Dorfer and G. Widmer. Training general-purpose audio tagging networks with noisy labels and iterative self-verification. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE2018)*, Surrey, UK, 2018.

M. Dorfer, R. Donner, and G. Langs. Constructing an un-biased whole body atlas from clinical imaging data by fragment bundling. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 219–226, Nagoya, Japan, 2013.

M. Dorfer, A. Arzt, and G. Widmer. Live score following on sheet music images. In *the Late Braking Demo at the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016a.

M. Dorfer, A. Arzt, and G. Widmer. Towards score following in sheet music images. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 789–795, New York City, USA, 2016b.

M. Dorfer, T. Kazmar, M. Smíd, S. Sing, J. Kneißl, S. Keller, O. Debeir, B. Luber, and J. Mattes. Associating approximate paths and temporal sequences of noisy detections: Application to the recovery of spatio-temporal cancer cell trajectories. *Medical Image Analysis*, 27:72–83, 2016c.

M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, USA, 2016d.

M. Dorfer, A. Arzt, and G. Widmer. Learning audio-sheet music correspondences for score identification and offline alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 115–122, Suzhou, China, 2017a.

M. Dorfer, J. j. Hajič, and G. Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, Kyoto, Japan, 2017b.

M. Dorfer, J. j. Hajič, A. Arzt, H. Frostel, and G. Widmer. Learning audio—sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval*, 1(1): 22–33, 2018a. doi:http://doi.org/10.5334/tismir.12.

M. Dorfer, J. j. Hajič, and G. Widmer. Attention as a perspective for learning tempo-invariant audio queries. In *ICML 2018 Joint Workshop on Machine Learning for Music*, Stockholm, Sweden, 2018b.

M. Dorfer, F. Henkel, and G. Widmer. Learning to listen, read, and follow: Score following as a reinforcement learning game (**Best Paper and Best Poster Award**). In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018c.

M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer. End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval (IJMIR)*, 7(2):117–128, 2018d.

A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 766–774, Montreal, Canada, 2014.

Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 1329–1338, New York City, USA, 2016.

Z. Duan and B. Pardo. A state space model for on-line polyphonic audio-score alignment. In *In Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.

J. W. Dunn, D. Byrd, M. Notess, J. Riley, and R. Scherle. Variations2: Retrieving and using music in an academic setting. *Communications of the ACM, Special Issue: Music information retrieval*, 49(8):53–48, 2006. doi:10.1145/1145287.1145314.

H. Eghbal-zadeh, M. Dorfer, and G. Widmer. A cosine-distance based neural network for music artist recognition using raw i-vector features. In *19th International Conference on Digital Audio Effects (DAFx16)*, 2016.

H. Eghbal-zadeh, B. Lehner, M. Dorfer, and G. Widmer. A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO) 2017*, pages 2749–2753, Kos, Greece, 2017.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

C. Fremerey, M. Clausen, S. Ewert, and M. Müller. Sheet music-audio identification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 645–650, Kobe, Japan, 2009.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

J. H. Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

A.-J. Gallego and J. Calvo-Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017. doi:10.1016/j.eswa.2017.07.002.

H. Garn, M. Waser, M. Lechner, M. Dorfer, and D. Grossegger. Robust, automatic real-time monitoring of the time course of the individual alpha frequency in the time and frequency domain. In *In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2227–2231, San Diego, USA, 2012.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323, Fort Lauderdale, USA, 2011.

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1319–1327, Atlanta, USA, 2013.

M. Grachten, M. Gasser, A. Arzt, and G. Widmer. Automatic alignment of music performances with structural differences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 607–612, Curitiba, Brazil, 2013.

B. Graham. Spatially-sparse convolutional neural networks. *CoRR*, abs/1409.6070, 2014.

E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

J. Hajič jr and P. Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th International Conference on Document Analysis and Recognition (ICDAR)*, pages 39–46, New York, USA, 2017.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12): 2639–2664, 2004. doi:10.1162/0899766042321814.

J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2760–2767, Sydney, Australia, 2013.

K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456, Lille, France, 2015.

Ö. Izmirli and G. Sharma. Bridging printed music and audio through alignment using a mid-level score representation. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 61–66, Porto, Portugal, 2012.

A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, Boston, USA, 2015.

A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems (NIPS)*, pages 1889–1897, Montreal, Canada, 2014.

R. Kelz and G. Widmer. Learning to transcribe by ear. *arXiv preprint (arXiv:1805.11526)*, 2018.

R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer. On the potential of simple framewise approaches to piano transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 475–481, New York City, USA, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint (arXiv:1411.2539)*, 2014.

F. Krebs, S. Böck, M. Dorfer, and G. Widmer. Downbeat tracking using beat synchronous features with recurrent neural networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 129–135, New York City, USA, 2016.

M. Krenn, M. Dorfer, O. A. J. del Toro, H. Müller, B. H. Menze, M. Weber, A. Hanbury, and G. Langs. Creating a large-scale silver corpus from multiple algorithmic segmentations. In *Medical Computer Vision: Algorithms for Big Data - International Workshop (MCV)*, pages 103–115, Munich, Germany, 2015.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, Lake Tahoe, USA, 2012.

F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 261–266, Vienna, Austria, 2007.

C. C. S. Liem, E. Gómez, and M. Schedl. PHENICX: Innovating the classical music experience. In *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–4, Torino, Italy, 2015.

M. Lin, Q. Chen, and S. Yan. Network in network. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, 26(2):181–191, 2005. doi:10.1016/j.patrec.2004.09.014.

J. R. Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(2):179–191, 1985.

J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint (arXiv:1410.1090)*, 2014.

K. Mardia, J. Kent, and J. Bibby. *Multivariate analysis*. Probability and mathematical statistics. Academic Press, 1979.

B. McFee, E. J. Humphrey, and J. P. Bello. A software framework for musical data augmentation. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 248–254, Málaga, Spain, 2015.

M. S. Melenhorst, R. van der Sterren, A. Arzt, A. Martorell, and C. Liem. A tablet app to enrich the live and post-live experience of classical concerts. In *Proceedings of the 3rd International Workshop on Interactive Content Consumption co-located with ACM International Conference on Interactive Experiences for Television and Online Video (ACM TVX)*, Brussels, Belgium, 2015.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119, Lake Tahoe, USA, 2013.

M. Miron, J. J. Carabias-Orti, and J. Janer. Audio-to-score alignment at note level for orchestral recordings. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 125–130, Taipei, Taiwan, 2014.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi:10.1038/nature14236.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 1928–1937, New York City, USA, 2016.

M. Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015.

M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 288–295, London, Great Britain, 2005.

E. Nakamura, P. Cuvillier, A. Cont, N. Ono, and S. Sagayama. Autoregressive hidden semi-markov model of symbolic music performance for score following. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 392–398, Málaga, Spain, 2015.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

B. Niedermayer and G. Widmer. A multi-pass algorithm for accurate audio-to-score alignment. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 417–422, Utrecht, The Netherlands, 2010.

M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, Bhubaneswar, India, 2008.

T. Papadopoulo and M. I. Lourakis. Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications. In *Proceedings of the 6th European Conference on Computer Vision (ECCV)*, Dublin, Ireland, 2000.

J. C. Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. R. Lanckriet, R. Levy, and N. Vasconcelos. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):521–535, 2014. doi:10.1109/TPAMI.2013.142.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

M. Prockup, D. Grunberg, A. Hrybyk, and Y. E. Kim. Orchestral performance companion: Using real-time audio to score alignment. *IEEE Multimedia*, 20(2): 52–60, 2013. doi:10.1109/MMUL.2013.26.

L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.

M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in neural information processing systems (NIPS)*, pages 6078–6087, Long Beach, USA, 2017.

C. Raphael. Music Plus One and machine learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 21–28, Haifa, Israel, 2010.

A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012. doi:10.1007/s13735-012-0004-6.

S. E. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–58, Las Vegas, USA, 2016.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, Munich, Germany, 2015.

L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004. doi:10.1162/089976604773135104.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pages 318–362. Cambridge, USA, 1986.

S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 2016. doi:10.1109/TASLP.2016.2533858.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *Workshop of the International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

A. Stuhlsatz, J. Lippel, and T. Zielke. Feature extraction with deep neural networks by a generalized discriminant analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):596–608, 2012. doi:10.1109/TNNLS.2012.2183645.

I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1139–1147, Atlanta, USA, 2013.

R. S. Sutton and A. G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

K. Swersky, J. Snoek, and R. P. Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems (NIPS)*, pages 2004–2012, Lake Tahoe, USA, 2013.

V. Thomas, C. Fremerey, M. Müller, and M. Clausen. Linking sheet music and audio - challenges and new approaches. In *Multimodal Music Processing*, pages 1–22. 2012. doi:10.4230/DFU.Vol3.11041.1.

A. Vall, H. Eghbal-zadeh, M. Dorfer, M. Schedl, and G. Widmer. Music playlist continuation by learning from hand-curated examples and song features: Alleviating the cold-start problem for rare and out-of-set songs. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems (DLRS@RecSys)*, pages 46–54, 2017.

A. Vall, M. Dorfer, M. Schedl, and G. Widmer. A hybrid approach to music playlist continuation based on playlist-song membership. In *33rd Symposium on Applied Computing (SAC)*, pages 1374–1382, Pau, France, 2018.

E. van der Wel and K. Ullrich. Optical music recognition with convolutional sequence-to-sequence models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 731–737, Suzhou, China, 2017.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NIPS)*, pages 5998–6008. Long Beach, USA, 2017.

I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

R. Vogl, M. Dorfer, and P. Knees. Recurrent neural networks for drum transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 730–736, New York City, USA, 2016.

R. Vogl, M. Dorfer, and P. Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 201–205, New Orleans, USA, 2017a.

R. Vogl, M. Dorfer, G. Widmer, and P. Knees. Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 150–157, Suzhou, China, 2017b.

W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1083–1092, Lille, France, 2015a.

W. Wang, R. Arora, K. Livescu, and J. A. Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4590–4594, South Brisbane, Australia, 2015b.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

C. Wen, A. Rebelo, J. Zhang, and J. Cardoso. A new optical music recognition system based on combined neural network. *Pattern Recognition Letters*, 58:1–7, 2015. doi:10.1016/j.patrec.2015.02.002.

G. Widmer. Getting closer to the essence of music: The *Con Espressione* manifesto. *ACM TIST*, 8(2):19:1–19:13, 2017. doi:10.1145/2899004.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi:10.1007/BF00992696.

Y. Wu, E. Mansimov, S. Liao, R. B. Grosse, and J. Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *arXiv preprint (arXiv:1708.05144)*, 2017.

K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2048–2057, Lille, France, 2015.

F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3441–3450, Boston, USA, 2015.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2017.

J. Zhao, M. Mathieu, R. Goroshin, and Y. Lecun. Stacked what-where auto-encoders. In *Workshop of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2015.

Curriculum Vitae of the Author

Personal Data

Name: Matthias Dorfer

Education

04/2015–11/2018	PhD in Informatics at the Johannes-Kepler University Linz. *Thesis:* "Multimodal Deep Representation Learning and its Application to Audio and Sheet Music."
03/2011–03/2013	Master of Science in Medical Computer Science at the Vienna University of Technology. *Thesis:* "A Framework for Medical-Imaging-Fragment Based Whole Body Atlas Construction"
10/2007–01/2011	Bachelor of Science in Medical Computer Science at the Vienna University of Technology. *Thesis:* "Electrical Stimulation of the Human Spinal Cord: Electro-physiological Characteristics of Spinal Reflexes"

Experience

since 04/2015	University assistant, PhD student (Johannes Kepler University Linz, Department of Computational Perception)
10/2013–04/2015	Researcher in knowledge based computer vision (Software Competence Center Hagenberg GmbH)
03/2013–10/2013	Research assistant in medical image processing (Medical University of Vienna, Computational Image Analysis and Radiology Lab)
10/2011–06/2012	Developer of bio-signal processing algorithms (Austrian Institute of Technology)
10/2011–02/2012	Tutor for Basics of Digital Image Processing (Vienna University of Technology)
03/2007–08/2007	IT-Developer (Fabasoft AG, E-Gov Solutions)

Awards and Prizes

09/2018	Best Paper Award at the the 19th International Society for Music Information Retrieval Conference (ISMIR) in Paris.
09/2018	Best Poster Award at the the 19th International Society for Music Information Retrieval Conference (ISMIR) in Paris.
2007-2011	Performance scholarship of the Faculty of Informatics at the Vienna University of Technology (2007, 2008, 2009, 2010, 2011)
	Foundation Stipend of Vienna University of Technology

Scientific Services

Reviewer for international conferences and journals:

International Society for Music Information Retrieval Conference (2018)
International Journal of Multimedia Information Retrieval
Workshop on Detection and Classification of Acoustic Scenes and Events (2018)

Teaching

WT 2018/19	KV Special Topics: Reinforcement Learning
WT 2018/19	Probabilistic Models (Exercises)
ST 2018	Seminar in Intelligent Information Systems: Multimedia Information Retrieval
ST 2018	Project in Networks and Security
WT 2017/18	KV Special Topics: Reinforcement Learning
WT 2017/18	Probabilistic Models (Exercises)
ST 2017	Seminar in Intelligent Information Systems: Multimedia Information Retrieval
ST 2017	Project in Networks and Security

Publications

Publications forming the main chapters of the thesis (■), publications from side projects and collaborations next to or before the thesis work (●), workshop contributions and technical reports (○):

- ○ M. Dorfer and G. Widmer. Training general-purpose audio tagging networks with noisy labels and iterative self-verification. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE2018)*, Surrey, UK, 2018

- ○ M. Dorfer, J. j. Hajič, and G. Widmer. Attention as a perspective for learning tempo-invariant audio queries. In *ICML 2018 Joint Workshop on Machine Learning for Music*, Stockholm, Sweden, 2018b

- ■ M. Dorfer, F. Henkel, and G. Widmer. Learning to listen, read, and follow: Score following as a reinforcement learning game (**Best Paper and Best Poster Award**). In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018c.

- ■ M. Dorfer, J. j. Hajič, A. Arzt, H. Frostel, and G. Widmer. Learning audio—sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval*, 1 (1):22–33, 2018a. doi:http://doi.org/10.5334/tismir.12.

- ■ M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer. End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval (IJMIR)*, 7(2):117–128, 2018d.

- ● A. Vall, M. Dorfer, M. Schedl, and G. Widmer. A hybrid approach to music playlist continuation based on playlist-song membership. In *33rd Symposium on Applied Computing (SAC)*, pages 1374–1382, Pau, France, 2018.

- ■ M. Dorfer, J. j. Hajič, and G. Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, Kyoto, Japan, 2017b.

- ■ M. Dorfer, A. Arzt, and G. Widmer. Learning audio-sheet music correspondences for score identification and offline alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 115–122, Suzhou, China, 2017a.

- R. Vogl, M. Dorfer, G. Widmer, and P. Knees. Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 150–157, Suzhou, China, 2017b.

- H. Eghbal-zadeh, B. Lehner, M. Dorfer, and G. Widmer. A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO) 2017*, pages 2749–2753, Kos, Greece, 2017.

◦ A. Vall, H. Eghbal-zadeh, M. Dorfer, M. Schedl, and G. Widmer. Music playlist continuation by learning from hand-curated examples and song features: Alleviating the cold-start problem for rare and out-of-set songs. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems (DLRS@RecSys)*, pages 46–54, 2017.

- R. Vogl, M. Dorfer, and P. Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 201–205, New Orleans, USA, 2017a.

■ M. Dorfer, A. Arzt, and G. Widmer. Towards score following in sheet music images. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 789–795, New York City, USA, 2016b.

◦ M. Dorfer, A. Arzt, and G. Widmer. Live score following on sheet music images. In *the Late Braking Demo at the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016a.

- R. Vogl, M. Dorfer, and P. Knees. Recurrent neural networks for drum transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 730–736, New York City, USA, 2016.

- R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer. On the potential of simple framewise approaches to piano transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 475–481, New York City, USA, 2016.

- F. Krebs, S. Böck, M. Dorfer, and G. Widmer. Downbeat tracking using beat synchronous features with recurrent neural networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 129–135, New York City, USA, 2016.

- H. Eghbal-zadeh, M. Dorfer, and G. Widmer. A cosine-distance based neural network for music artist recognition using raw i-vector features. In *19th International Conference on Digital Audio Effects (DAFx16)*, 2016.

◦ M. Dorfer and G. Widmer. Towards deep and discriminative canonical correlation analysis. In *ICML 2016 Workshop on Multi-View Representation Learning*, New York, USA, 2016a.

■ M. Dorfer and G. Widmer. Towards end-to-end audio-sheet-music retrieval. In *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, Barcelona, Spain, 2016b.

■ M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, USA, 2016d.

- M. Dorfer and J. Mattes. Recursive water flow: A shape decomposition approach for cell clump splitting. In *13th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 811–815, Prague, Czech Republic, 2016.

- M. Dorfer, T. Kazmar, M. Smíd, S. Sing, J. Kneißl, S. Keller, O. Debeir, B. Luber, and J. Mattes. Associating approximate paths and temporal sequences of noisy detections: Application to the recovery of spatio-temporal cancer cell trajectories. *Medical Image Analysis*, 27:72–83, 2016c.

- M. Krenn, M. Dorfer, O. A. J. del Toro, H. Müller, B. H. Menze, M. Weber, A. Hanbury, and G. Langs. Creating a large-scale silver corpus from multiple algorithmic segmentations. In *Medical Computer Vision: Algorithms for Big Data - International Workshop (MCV)*, pages 103–115, Munich, Germany, 2015.

- M. Dorfer, R. Donner, and G. Langs. Constructing an un-biased whole body atlas from clinical imaging data by fragment bundling. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 219–226, Nagoya, Japan, 2013.

- H. Garn, M. Waser, M. Lechner, M. Dorfer, and D. Grossegger. Robust, automatic real-time monitoring of the time course of the individual alpha frequency in the time and frequency domain. In *In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2227–2231, San Diego, USA, 2012.