Technische Universität Berlin

Institut für Sprache und Kommunikation

Fachgebiet Audiokommunikation

Fakultät I Geisteswissenschaften

Masterarbeit

Transfer and multitask learning for music classification tasks

Sebastian Cycoń

[redacted]

19.12.2020

Erstgutachter

Prof. Dr. Stefan Weinzierl

Zweitgutachter

Dr. Athanasios Lykartsis

Abstract

Deep neural networks outperform conventional machine learning techniques on music information retrieval (MIR) tasks, but require larger datasets for the training. The availability of large dataset though, especially in the music domain, is very rare. Moreover, the available datasets are of a low quality. Recent research tried to face this problem by transferring knowledge and sharing trainable model parameters for multiple datasets and related tasks. Transfer learning enables tasks to benefit from systems trained on large datasets. If the tasks share similar low-level representations, features extracted from those system can help new, related tasks. Another method is multitask learning. When training jointly multiple similar tasks, shared feature representations can benefit from the availability of additional data. Simultaneously, with both approaches, the capacity of deep learning networks is reduced facilitating them to detect useful features even with small available data.

This master's thesis expands on these approaches of transfer learning and multitask learning, which proved to be beneficial when working with smaller datasets. For the purposes of this thesis, a new, relatively small dataset including songs associated with tags for DJ application was manually created from scratch. A system trained on automatic song tagging using a large dataset served as the source task for transfer learning. The applicability of the system for various music classification tasks was evaluated. Training in singletask as well as various multitask settings, this work reveals advantages and limitations of solving different classification tasks in a multitask setting. Following current studies in computer vision, a

novel approach of combining transfer with multitask learning was used to classify the songs on multiple label categories. With adding data augmentation methods and increasing the data sample size, the impact of transfer and multitask learning on the performances of the networks was put into a wider context, additionally indicating the required size of training data to achieve satisfying results. This research is expected to enable challenging tagging tasks to be solved when only a relatively small amount of data is available due to high creation costs or difficulties in the annotation.

Zusammenfassung

Neuronale Netze übertreffen herkömmliche Techniken des maschinellen Lernens bei MIR-Aufgaben (Music Information Retrieval), erfordern jedoch größere Datensätze für das Training. Die Verfügbarkeit großer Datenmengen, insbesondere im Musikbereich, ist jedoch sehr selten. Darüber hinaus sind die verfügbaren Datensätze von geringer Qualität. Forschungen haben versucht, diesem Problem entgegenzutreten, indem sie Wissen übertragen und trainierbare Modellparameter für mehrere Datensätze und verwandte Aufgaben gemeinsam nutzen. Mit Transfer Learning können Klassifizierungsprobleme von Systemen profitieren, die auf großen Datenmengen trainiert sind. Wenn diese Probleme ähnliche Merkmale auf niedriger Ebene aufweisen, können aus diesen Systemen extrahierte Merkmale beim Lösen neuer Aufgaben helfen. Eine andere Methode ist das Multitask Learning. Wenn mehrere ähnliche Aufgaben gemeinsam trainiert werden, können gemeinsam genutzte Feature-Darstellungen von der Verfügbarkeit zusätzlicher Daten profitieren. Gleichzeitig wird bei beiden Ansätzen die Kapazität von Deep-Learning-Netzwerken verringert, sodass sie auch bei kleinen verfügbaren Daten nützliche Funktionen erkennen können.

Diese Masterarbeit erweitert diese Ansätze des Transfer- und Multitask-Lernens, die sich bei der Arbeit mit kleineren Datensätzen als vorteilhaft erwiesen haben. Für die Zwecke dieser Arbeit wurde ein neuer, relativ kleiner Datensatz mit Songs, die mit Tags für DJ-Anwendungen verknüpft sind, von Grund auf neu erstellt. Ein System, das auf das automatische Taggen von Songs unter Verwendung eines großen Datensatzes trainiert wurde, wurde für das Transfer Learning benutzt. Die Anwendbarkeit des Systems für verschiedene Musikklassifizierungsaufgaben wurde bewertet. Beim Training in Singletask- sowie verschiedenen Multitask-

Umgebungen wurden die Vor- und Nachteile verschiedener Klassifizierungsaufgaben in einer Multitask-Umgebung aufgezeigt. Inspiriert von aktuellen Studien im Feld der Bildklasifizierung wurde ein neuartiger Ansatz zur Kombination von Transfer mit Multitask Learning verwendet, um die Songs in mehrere Label-Kategorien einzuteilen. Mit dem Hinzufügen von Datenaugmentierungsmethoden und dem damit verbundenen Erhöhen des Datensatzes wurde der Einfluss von Transfer und Multitask-Lernen auf die Leistung der Netzwerke in einen breiteren Kontext gestellt und zusätzlich die erforderliche Größe der Trainingsdaten angegeben, um zufriedenstellende Ergebnisse zu erzielen. Es wird erwartet, dass diese Forschung die Lösung herausfordernder Tagging-Aufgaben ermöglicht, wenn aufgrund von Schwierigkeiten bei der Erstellung nur eine relativ kleine Datenmenge verfügbar ist.

Contents

List of Figures ix

List of Tables xi

List of Abbreviations xiii

1 Introduction 1

2 State of the Art 7

3 Datasets 11

3.1 Target Task: DJ-Tags Dataset 11

3.2 Source Tasks: DJCity Datasets 14

3.3 Source Task: Million Song Dataset 17

4 Method 19

4.1 Convolutional Neural Networks 19

4.2 Transfer Learning 20

4.3 Multitask Learning 23

4.4 Data Augmentation 25

5 Experimental Setup & Results 29

5.1 Overview 29

5.2 Experiment 1: CNN with DJCity Datasets 34
5.2.1 Inputs & Architecture 34
5.2.2 Results 36
5.3 Experiment 2: MLP with MSD Features 40
5.3.1 Inputs & Architecture 40
5.3.2 Results 43

6 Discussion **47**

7 Conclusion **63**

Bibliography **I**

Appendix **V**

List of Figures

4.1 Proposed model for transfer learning with DJCity datasets 21
4.2 Overview of the feature extractor designed by Choi et al. [1] 22
4.3 Proposed model design for multitask learning 24
4.4 Proposed model for multitask learning with shared layers 24
4.5 Proposed model for multitask learning with outputs of genre & energy branch as additional features for the subgenre & situation task 25
4.6 Mel-spectrogram of the song "Avicii ft. Aloe Blacc - S.O.S" (a) pitch shifted one tone down (b) pitch shifted one tone up 26

6.1 Comparison of the performances of the proposed transfer learning models in a basic multitask learning setting 48
6.2 Comparison of the performances of the proposed multitask learning models 49
6.3 Accuracy scores of the main genre, subgenre & intro-version predictions for the proposed models 51
6.4 Confusion matrices for the intro version prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation 53

6.5 Confusion matrices for the main genre prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation 55

6.6 Confusion matrices for the subgenre prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selection, (d) MLP with MSD multitask Output-Feature model with data augmentation 57

6.7 (a) R2 Scores of the energy prediction and (b) AUC scores of the situation prediction 58

6.8 Scatter plots for the energy prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation . . . 59

List of Tables

3.1 Overview of the DJ-Tags dataset with a total number of 449 Songs 15

3.2 Overview of the DJCity Datasets 17

3.3 Overview of the Million Song Dataset 18

5.1 Total validation loss for the MTL model in regard of the song positions features are extracted from 30

5.2 Prior knowledge of the given tasks 32

5.3 Baseline CNN model for genre classification 33

5.4 Results for the CNN singletask model without and with augmented dataset 37

5.5 Results for the proposed mutltiask models 37

5.6 Results for the source tasks 38

5.7 Results for the transfer learning methods. *Combined* stands for the pre-trained model with shared layers and data augmentated input . 39

5.8 Feature selection algorithms used for the tasks. Values in parentheses correspond to multitask settings 41

5.9 MSD transfer learning model for energy classification 41

5.10 Results for the proposed MLP networks with the transfer learning system trained on MSD 44

5.11 Results for additionally pre-training MLP models with the DJ City datasets in accuracy. Values in parentheses correspond to training with augmented input data 45
5.12 Results for further multitask methods to increase performance . . . 45
5.13 Overview of the best performing MLP networks taking augmented data without feature selection as input 46

6.1 Results for the situation prediction task with the baseline CNN model (AUC=0.62) 60
6.2 Results for the situation prediction task with the CNN multitask model with shared layers and data augmentation (AUC=0.75) . . . 60
6.3 Results for the situation prediction task with the MLP with MSD singletask model with feature selection (AUC=0.77) 61
6.4 Results for the situation prediction task with the MLP with MSD multitask Output-Feature model with data augmentation (AUC=0.88) 61
6.5 Comparison of computational costs 62

List of Abbreviations

ACC	Accuracy
API	Application Programming Interface
AUC	Area Under the Receiver Operating CharacteristicCurve
CNN	Convolutional Neural Networks
CONV	Convolutional Layer
DJ	Disc Jockey
MI	Mutual Information
MIR	Music Information Retrieval
MLP	Multilayer-Perceptron
MSD	Million Song Dataset
MSE	Mean Squared Error
MTAT	MagnaTagATune Dataset
MTL	Multitask Larning
RFECV	Recursive Feature Eimination with Cross-Validated Selection
TL	Transfer Learning

1 Introduction

Music information retrieval (MIR) gained its popularity from the digitization of music which resulted in a wide access to music itself. Along with machine learning algorithms, MIR allows music streaming platforms such as Spotify[1] or Youtube[2] to catalogue music, create playlists, or provide appropriate recommendations. Tzanetakis et al. [2] use audio feature engineering coupled with conventional machine learning techniques, requiring hand-designed features for music genre classification, which is a common task in MIR. Since the genre is only one semantic category to describe a song, more recently a lot of research has been invested in automatic tagging of music. Auto-tagging tasks use large datasets created with social tagging. These datasets include tags related not only to the genre but also mood, instrumentation, era, and more and therefore provide richer descriptions of songs. However, the complexity of music in general as well as the multilabel problem nature of auto-tagging in contrast to genre classification as a singlelabel problem makes the use of conventional machine learning extremely challenging. In singlelabel classifications the classes are mutually exclusive. Whereas for multilabel problems one data sample can be associated to multiple labels increasing the number of cases exponentially in comparison to singlelabel problems [3]. In recent years, deep learning, especially convolutional neural networks (CNN), clearly outperformed conventional machine learning approaches [3] [4] and became very

[1] https://www.spotify.com
[2] https://www.youtube.com

popular for MIR tasks. Since deep models need a lot of training data, large datasets, like the Million Song Dataset (MSD) [5] or MagnaTagATune [6], are used for auto-tagging tasks. MSD is the biggest available dataset of metadata for one million popular songs. Linked to social tags from Last.Fm[3], MSD provides tags for around five-hundred thousand songs [7].

However, since large datasets like MSD mostly consist of user-generated tags, they are unstructured and noisy, have no constrained vocabulary, and therefore highly correlated tags as well as missing labels. Datasets with annotations by music expert annotators are more reliable and provide an overall higher quality. Due to the high costs of creation, datasets with professional annotations are rare and either relatively small or specific to one task. An example of this is the extended ballroom dataset for ballroom dance genre classification that includes 4180 samples [8]. Nonetheless there are approaches to create large and high-quality datasets by extracting groups of tags from large datasets coming from social tagging with analysis of tag similarity and verification of the data with additional datasets [9] [10]. To face the problem of missing large and qualitative datasets a different approach is recently gaining popularity: Transfer learning.

Using transfer learning, a model is created that learns low dimensional representations on a source task and use the trained features as input for other related tasks. In deep neural networks, the features in the first layers are more generic whereas the later layers become specific to the task. Since similar tasks share low-level features, the advantage of transfer learning gives the possibility to train these first layers on related tasks with large available datasets. By using the learned features as the input for a target task, the last specific layers are then fine-tuned via

[3] https://www.last.fm

the data generated by the target task [11]. Similar to transfer learning, multitask learning can improve performances with learning simultaneously a jointly shared space between different but related tasks by transferring knowledge [12]. Multitask learning allows tasks with small available data (e.g., data that is difficult to annotate) to make use of available data of related tasks for the training process. In theory, both approaches have the potential to increase performances when only small datasets are available, since the amount of training data to adjust parameters in deep networks will be increased. In transfer learning the data of the related source task is used for training and in multitask learning all of the data from the multiple tasks are used for aligning the networks' parameters. This can result in more accurate models. And since the actual number of trainable parameters is being reduced, a more efficient network that is less prone to overfitting is achieved [13].

Following studies on transfer learning and multitask learning in MIR, this master's thesis presents a novel approach of combining both concepts. With these methods, several deep learning networks are build to classify songs on a new, manually created dataset. The relatively small dataset includes song tags for DJ application. DJs tag their songs to improve their library organization, playlist creation and performances. Especially for open format DJs, working at dissimilar events and therefore owning large music libraries including a wide range of music styles, proper song tagging is an essential part of their gig preparation. The dataset used for the target tasks in this thesis includes song tags for the four categories: genre, energy level, performance situation and intro version. The possibility of training these tasks simultaneously despite being associated with different classification types (multiclass, multilabel and regression problems) and the consequent problematics is demonstrated. The categories having different numbers of classes

and some categories consisting of balanced and imbalanced data result in more and less difficult tasks to learn. A positive effect of easier tasks on more complex tasks trained jointly in a multitask setting is evaluated in this thesis. Two different approaches of transfer-learning are implemented. First, additional datasets with tasks similar to the target classes are created. Auto-tagging on MSD serves as the second approach for the source task of transfer learning. Comparing the benefits of both source task approaches for the target dataset give an insight into the applicability of the auto-tagging task for various music classification tasks.

The research goal is to investigate the possible advantages of multitask and transfer learning when working with small datasets, understanding the process of adapting deep models to specific MIR tasks with small available data, and examining the differences of singletask versus multitask learning in regard to the model's performance and efficiency. Multitask learning works especially well for learning jointly low-level features [14]. Since the transfer learning process already provides low-level features trained on a large dataset it is interesting to see if there is potential for any improvement with creating an additional shared learning space for the target tasks. Additionally, data augmentation is implemented as another tool to handle small datasets. This puts the benefit of transfer and multitask learning into context and demonstrates the minimum required size of a dataset to achieve satisfying results. The motivation for creating a new dataset lies in having more and less complex data with predefined tags and tag categories ready for a multitask classification as well as the ability to examine the applicability of transfer learning for new, yet unconsidered tasks. So far, the target task on transfer learning approaches in MIR research mainly contains genre classification (e.g., [15]). Since MSD as the source dataset mostly has tags related to genre, mood, and instrumentation [9], it is interesting to see how the classification of not closely

related tags, like energy level or situation, will perform.

Working as a DJ myself, I am aware of the high time investment that tagging music libraries requires. This research has the potential to help accelerate this process and could therefore be of interest for DJs, who emphasize tagging their music. It can find utilization for record pools as well, which in the future could provide songs for DJs including useful metadata. DJ software, like Rekordbox[4], provides the ability for DJs to tag their songs inside of the DJ software to organize the music in playlists. In Rekordbox the default selectable tags are sorted in the categories genre, components and situation. It might as well be of interest to automatically tag songs inside the software. In general, this research on transfer and multitask learning is of interest for specific music classification tasks with no large datasets available, unlike auto-tagging tasks or common classification tasks like genre classification.

[4]https://rekordbox.com

2 State of the Art

Due to the complexity of music in general, automatic music classification is a complex task. With deep neural networks (DNN) becoming popular in the machine learning field, new possibilities for music classification were introduced. For conventional machine learning, audio features had to be created with signal processing techniques. But selecting task-relevant features turned out to be very difficult and captured a large part in implementing systems for music classification problems. With DNNs undertaking this problem by creating feature representations specific to the tasks, they became very popular in the field of music information retrieval (MIR). The drawback is that a lot of data is required for deep learning models to extract the feature representations from the input data. Therefore, auto-tagging tasks take advantage of large available datasets, like the Million Song Database (MSD) or MagnaTagATune (MTAT).

After being introduced to the MIR field [16] convolutional neural networks (CNN), technologies originally designed for the field of image information retrieval were applied to several auto-tagging systems ([3, [4], [17]). In [3] CNN models of different sizes are presented to predict tags on MSD and MTAT. More complex models benefit most from the availability of large datasets, compared to smaller models achieving better results with less available training data.

After showing to be an effective method in computer vision with ImageNet classification [18] being used as the source task for other image classification tasks [19], transfer learning is also beginning to be used in MIR tasks. ImageNet classification appears to be suitable as a source task in computer vision due to its high number of images and categories. Equivalently, Oord et al. [20] introduce the approach of using features learned on MSD to predict other music classification problems. A multilayer-perceptron (MLP) is implemented, taking features extracted from the source model as the input. Due to the resulting reduced complexity of the networks, they are able to classify target tasks, in this case genre and tag prediction, with smaller available training data. Compared to user listening preference prediction, the transfer learning approach achieves better results with tag prediction as the source task. Furthermore, transfer learning turns out to be particularly advantageous, if the target task is similar to the source task. Since MSD is the biggest available dataset and the model is trained on a wide range of semantic tasks including genre, instrumentation, mood, etc. Choi et al. [1] recommend auto-tagging with MSD as the source task for other music classification problems as well. Adopting this transfer learning approach, several systems based on CNNs are trained on MSD [1] [21]. Choi et al. [1] present a CNN consisting of five convolutions as the source system for transfer learning. The features from the trained CNN model are aggregated and used as an input for other tasks including genre classification, vocal/non-vocal classification, emotion prediction, speech/music classification, and acoustic event classification which lead to achieving overall good results. The evaluation of the performances however makes evident, that target tasks which are closely related to the tags in MSD, like genre or vocal/non-vocal classification, and perform better less related tasks, like acoustic event classification. Ghosal et al. [15] approve the suitability of this system for transfer learning. In comparison to other implemented CNN models, a

simple MLP network using aggregated features from the pre-trained CNN system as inputs, achieves better results for solving genre classification problems. In this master's thesis, this pre-trained CNN will be used as the source system for transfer learning as well. Unlike most transfer learning approaches predicting closely related tasks, like tag and genre prediction, the system will be used to predict DJ-related tasks that are less related to the MSD tags.

Similar to transfer learning, the concept of multitask learning is recently getting more attention in the MIR field. Parallel to the transfer learning approach, where the tasks take advantage of aggregated audio features from other datasets, a shared embedding feature space can be implemented in a multitask setting, taking advantage of the semantic similarities of the tasks. Exploring this approach for music annotation, Weston et al. [14] record an improved performance of each task and additionally get faster and more efficient training process. Böck et al. [13] take advantage of multitask learning for tempo estimation and beat tracking with improving one task by learning from the other. Since high-quality annotated data for beat tracking is rare, it is particularly benefitting from the availability of a related task and data. Furthermore, the smaller number of weigths that have to be trained leads to a more efficient model decreasing the risk of overfitting. Schindler et al. [7] successfully apply the multitask learning approach on auto-tagging. After deviding the tag labels on MSD into the four categories genre, style, mood and theme, the multitask approach leads to an improvement for each label group. Solely the genre classification problem is not showing any improvement when learning jointly with the other tasks.

In the computer vision domain, studies combine these two concepts of transferring information. Zhang et al. [22] and Samala et al. [23] use large natural

image datasets as source tasks to provide information for classifying biological images. Pre-trained on ImageNet data CNNs are fine-tuned in a multitask setting. The models perform better using multitask learning in comparison to singletask transfer learning approaches. However, to the best of the author's knowledge, no studies are combining both concepts in the MIR field yet.

3 Datasets

3.1 Target Task: DJ-Tags Dataset

As stated in the introduction, a new dataset has been created to examine the behaviour of transfer and multitask learning for multiple classification tasks. The goal was to create a high qualitative dataset including music classification tags similar as well as dissimilar to the MSD tags. These tags were divided into multiple classification problems (multiclass, multilabel and regression). The dataset consists of songs from a personal DJ-music library. To ensure a qualitative dataset, a constrained vocabulary for the tags has been defined. The songs were manually tagged by the author according to the defined DJ-related tags inside the categories genre (disco, pop, etc.), energy-Level (on a scale from 1 to 4), situation (lounge, warmup, etc.) and intro-version (instrumental intro, acapella intro, etc.). Tagging was accomplished inside the music management software Mediamonkey[1], where the annotations were written in the audio's metadata.

The biggest obstacle was to achieve balanced data in all classification groups. With the subsampling method, the songs were picked carefully in a way to ensure an as uniform as possible distribution of the tags. The subsampling method describes the process of reducing the sample size of classes with high count to

[1] https://www.mediamonkey.com/

the number of samples of the smallest class. Co-occurrences of labels across the groups were observed. For example, songs with the genres disco and soul mostly were labelled as lounge in the situation group. This was also considered during subsampling, ensuring a wider distribution of labels across the categories to prevent the later proposed networks to just learn co-occurrencies. Another method was consolidating similar genres to higher level classes resulting in the class categoriy named main genres. To differentiate between the genre groups, the actual genres are named subgenres. The idea behind this method was to create a more balanced class category with a smaller number of classes resulting in the later proposed networks being expected to be trained easier by this task. Therefore, it will be possible to compare the training from rather easy tasks to difficult tasks. Additionally, the potential to improve different and complicated tasks in a multitask setup can be evaluated.

The subgenres are directly associated to the main genres. Most of the terms for the main genre and subgenre lables should be familiar and therefore will not be further explained. I want to note though, that the genre latin dance consists of several latin american ballroom dance styles like salsa, rumba or bachata. Afrobeats, not a latin genre per se, is associated with the main genre latin due to similar characteristics to dancehall. Pop and rock, although quite dissimilar genres, are merged to one main genre to ensure uniformly distributed classes.

A large part of DJ music include intros added to the beginning of the songs to make it easier for DJs to mix from one track to another. A few extracted bars of the track's instrumental, often the instrumental of the chorus, most commonly serve for intros, named as Intro. Mostly found in electronic music, like house music, extended mixes of the songs include additional instrumental elements in the beginning, named as Extended-In. Unlike the already mentioned intros, those

instrumental elements usually start with just a few looped elements which are constantly surrounded by more instrumental elements every few bars. Further intro versions are Aca-Ins and Clapapella-Ins, describing non instrumental intros consisting only of vocal melodies and vocals in combination with clap sounds appearing every one or two beats, respectively. No Intro describes songs without an additional intro in the beginning.

As can be assumed by the term, the situation tags describes the time and place of DJ performances a song fits in. Lounge describing background music, summer describing music fitting in an outdoor warm weather setup, warmup describing danceable but either less energetic or less popular music than music labelled as midnight, describing danceable music for regular dance floor settings. Builddown corresponds to less energetic or more sentimental music usually played at the ending of the dance floor setting. In contrast to the described labels focusing on the playtime of an event, the remaining labels serve for further classification of electronic music considering the venue in which the music was designed for, with smallroom, mediumroom and bigroom corresponding to small underground clubs, larger, commercial club and big festival stages, respectively. Situation is an exceptional category in this dataset. On the contrary to the other categories, multiple tags can be applied to any song, making the situation prediction a multilabel problem.

The tag category energy distinguish the songs into four levels of energy with 100 being the lowest and 400 the highest energy level. High numbers were chosen to avoid conflicts with the songs' track numbers.

Due to the high subjectivity of tagging and to ensure the datasets reliability, it

was jointly created by the author and an additional DJ. The additional DJ was instructed to verify the song tags with individually annotating the songs tags. To minimize the number of tags, especially tags of small count, the annotator was instructed to use the provided vocabulary. Interestingly, the intro version tags were the only tags remained unchanged, implying a high objectivity of this label class. Overall 30 subgenres, 38 situation and 56 energy level tags were changed by the annotator, indicating a particularly high subjectivity for assessing energy levels. Eventually, the dataset was updated with the annotators changes. Table 3.1 gives an overview of the resulting dataset consisting of a total number of 449 songs.

The genre data is highly balanced, while subgenre labels' counts range from 9 for tech house to 31 for rap. The data in intro version is rather imbalanced as well, with counts ranging from 68 for Extended-In to 111 for Intro. The energy task is relatively balanced for the first three energy-levels while the highest level consists of only less than a half of the counts the three other levels consist of. The situation task is highly imbalanced with tag counts ranging from 47 for bigroom to 198 for midnight.

3.2 Source Tasks: DJCity Datasets

The following datasets were created to serve as source tasks for transfer learning. The audios for the datasets derive from the music platform DJCity[2]. DJCity is a digital record pool designed for DJs. The record pool offers a large number of todays popular, mostly danceable and club-ready music. The music comes in different versions, remixes, edits and intros. Every song is tagged with genre-related tags. During a personal conversation, one of DJCity's employee working as a music

[2]https://www.djcity.com/

Task	Problem type	#classes	Labels (#appearances)
Main genre	Multiclass	6	Bass (74), Hip Hop (74), House (76), Latin (76), Pop/Rock (73), Soul (76)
Subgenre	Multiclass	23	Bass Beats (20), EDM Trap (27), Future Bass (16), Twerk (11); R&B (26), Rap (31), Trap (17); Deep House (18), Disco House/ Nu Disco (12), Electro House (19), Progressive House (18), Tech House (9); Afrobeats (24), Dancehall (15), Latin Dance (16), Reggae (11), Reggaeton (10); Electro-Pop (21), Pop (29), Rock/Alternative (23); Disco (27), Funk (23), Soul (26)
Intro version	Multiclass	5	Aca-In (90), Clapapella-In (76), Extended-In (68), Intro (111), No Intro (104)
Energy	Regression	N/A	100 (125), 200 (139), 300 (128), 400 (57)
Situation	Multilabel	8	Lounge (103), Summer (78), Warmup (94), Midnight (198), Builddown (72); Smallroom (53), Mediumroom (49), Bigroom (47)

Table 3.1: Overview of the DJ-Tags dataset with a total number of 449 Songs

editor, outlined the company's music tagging process. Every newly added song is being described with a genre tag, e.g. "Hip Hop". Additionally, another tag for further description of the song is added, e.g. "German Rap". In electronic music it is common to add even more tags describing the genre and style. Some of the songs also receive tags related to a special DJ-Edit.

The whole catalogue of the record pool was downloaded in November 2019. The total amount of songs counts up to around 78,000 with 152 available tags. A lot of

songs are multiple copies, due to being aligned to more than one tag. The motivation was to create three datasets with similar labels to the target datasets labels. Since most of the tags are genre-related, datasets similar to the genre-related tags of the target dataset were created. Songs labeled with the same or similar tags as the target datasets tags were brought in consideration. The tag distribution of the downloaded catalogue was very uneven with only one song for merengue electronico and around 14,000 songs for hip hop. Subsampling the dataset a more even distribution of the tags was achieved, resulting in a subgenre dataset, including 18 of the 23 subgenres presented in the target dataset. A few music styles were missing, like electro-pop. Soul, disco and funk were merged to one genre due to a low individual number of song counts.

According to the creation for the target dataset, the genres were merged to main genres. A lot of the songs are offered in different versions. Therefore for genres with particularly high song count only one version per song aligned to the genre was selected. To ensure quality of the dataset, it was also considered to select only songs which are labelled solely to one genre. This resulted in a minimum of 159 songs for most subgenres and a minimum of 1320 for most main genres corresponding to the second smallest count for the main genre. Genres with higher available counts were randomly subsampled to this number to achieve a balance in the datasets. Some of the genre tags remained underrepresented in the dataset, like the main genre class soul, consisting of only 97 songs tagged as soul, disco or funk as well as subgenre class progressive house (60 songs).

A third dataset was created according to the target dataset's intro version task. A lot of the music provided by DJCity include different intro versions for the individual tracks, especially the previously described Intro, Aca-In, and No Intro

Task	#songs	#classes	Labels (#appearances)
Main genre	6697	6	Bass (1320), Hip Hop (1320), House (1320), Latin (1320), Pop/Rock (1320), Soul (97)
Subgenre	2635	18	Trap (159), Future Bass (154), Twerk (155); R&B (159), Hip Hop (159): Deep House (159), Disco House/ Nu Disco (107), Electro House (159), Progressive House (60), Tech House (159); Afrobeats (159), Dancehall (159), Latin-Dance (159), Reggae (159), Reggaeton (159); Pop (154), Rock/Alternative (159); Soul (97)
Intro version	369	5	Aca-In (123), Intro (123), No Intro (123)

Table 3.2: Overview of the DJCity Datasets

versions. Songs, that are represented in every of the three different intro versions, were considered for the dataset, resulting in 369 songs (129 songs with 3 versions). Even though this dataset, unlikely to general transfer learning approaches, is smaller than the target dataset, it is expected to be easier for the later proposed networks to learn the musical patterns belonging to this task, while the presented songs only differ in the intro version. The resulting datasets derived from DJCity are reported in table 3.2.

3.3 Source Task: Million Song Dataset

Being the largest available dataset, MSD is one of the most popular datasets in the auto-tagging field. Connected to the Last.Fm API[3], it provides crowd-sourced music tags for 505,216 songs. This thesis follows the approach of transfer learning with auto-tagging on MSD as the source task presented by Choi et al. [1].

[3]https://www.last.fm/

Problem type	#songs	Categories (#labels)
Multilabel	242,842	Genre (28), Era (5) Instrumentation (5), Mood (12)

Table 3.3: Overview of the Million Song Dataset

Accordingly, to minimize the number of tags with small counts, the training was limited to the 50 most popular tags, resulting in 242,842 audio samples associated with at least one of the 50 tags. Those tags include 28 genres (rock, pop etc.), 5 eras ('60s'-'00s'), 5 instrumentations (guitar, female voice, etc) and 12 moods (happy, sad, etc) with a distribution lying between 1,257 for the least tag counts (happy) and 52,944 for the tag with the most occurrences (rock). Table 3.3 gives an overview of the dataset.

The social tags are weakly labelled and noisy, including incorrect annotations and diverse text, since every listener is able to use any terms to label the songs. Additionally, the dataset is highly imbalanced. Yet, Choi et al. [1] provide a system trained on MSD and recommend its usage for transfer learning for various music classification tasks. The advantage of the system is clearly the large number of samples it is trained on. The system provides low- and high-level feature representations for learning and due to the diverse semantic music annotations and different tag categories included in MSD, especially the low-level features can be very interesting as well for dissimilar music classification tasks.

4 Method

4.1 Convolutional Neural Networks

One of the main outcomes of this work is to evaluate the applicability of the auto-tagging task on large, crowd-sourced datasets as the source task for transfer learning on various music classification tasks. To do so, two different transfer learning approaches are proposed. First, the deep learning model supposed to be used for classifying the target tasks with the DJ-Tags dataset will be trained by similar tasks with a higher number of available data. For this approach, the DJCity datasets are used as the source tasks for transfer learning. In the second experiment auto-tagging on MSD will serve as the source task for transfer learning. The learned features trained on the large dataset will be aggregated and used as the input for new multiple classification tasks.

CNNs provide the basis for the deep learning architectures in both experiments. Choi et al. [3] recommend CNNs for music classification tasks, especially for music tagging, since one of the main advantages of CNNs is the ability to detect data patterns that are locally invariant like musical events appearing at different locations in the time-frequency domain. CNNs basically consist of the functional interaction of two types of layers, the convolutional and the pooling layers. After every convolutional and pooling layer semantically similar data patterns are com-

bined, model complexity is reduced and with each layer more complex features, detected in the previous layer, can be combined, leading to high-level features. As the tasks in the proposed dataset, like the genre or situation classification tasks, are supposed to be rather characterized by high-level features, the training benefits from this hierarchical learning property of CNNs. It is interesting to evaluate the suitability of CNNs for the intro version task, which labels being strongly dependent on musical patterns in the beginning of the tracks. CNNs are chosen over other variants like convolutional recurrent neural networks, because they provide comparable results and a more stable and computational faster training [24].

4.2 Transfer Learning

Transfer learning describes the process of taking benefit of related tasks trained with larger datasets, referred to as source tasks. Transferring the existing knowledge from training the source tasks to the so called target tasks, that have only small available data for training, can improve the performances of the target tasks. Related tasks share similar low-level feature representations. Especially tasks being rather described by high-level features, as it is the case in the proposed target tasks. Just in the higher level representations, the features become specific to the task. To take advantage of transfer learning, one possibility is to train a network on a source task with a large dataset. Afterwards, this pre-trained network is further trained, usually with a smaller learning rate, on the target task. With this method, often referred to as fine-tuning, the trainable parameters are adjusted either on the whole network or only on higher level feature representations. Additionally, the reduced complexity of the network results in a more efficient and faster training and is less prone to overfit. Figure 4.1 illustrates this transfer learning approach that will be implemented in the first experiment. The illustrated model

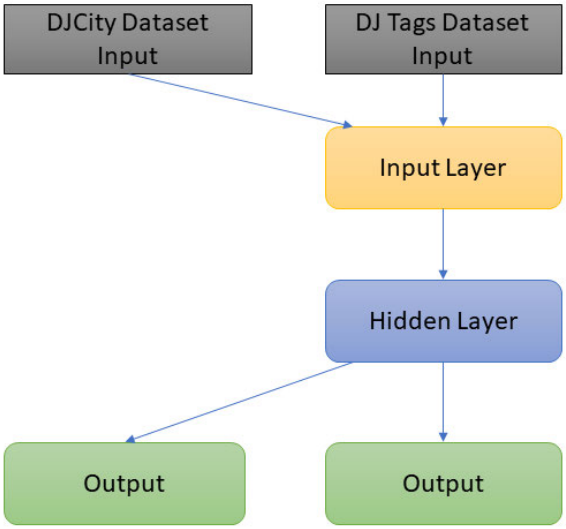


Figure 4.1: Proposed model for transfer learning with DJCity datasets

corresponds to a singletask CNN model with the hidden layer corresponding to multiple convolutional and pooling layers. The model will be trained by the DJCity datasets, serving as the source tasks. Since the genre task has the largest data of the source tasks, it will train the network's parameters first. The first layer's parameters will be then transferred to the subgenre and intro version networks. These networks are trained by the remaining source tasks (DJCity subgenre & DJCity intro version), while already taking advantage from the pre-trained low-level feature space. Finally the model's parameters will be fine-tuned with the proposed DJ-Tag dataset to classify the target tasks.

Another method taking advantage of transfer learning is to build a classifier on top of a pre-trained network. With the pre-trained network, features are aggregated from the new, smaller dataset before being used as the input data for a target classifier. This method is utilized in the second experiment. Experiment 2 is based on the proposed CNN system for transfer learning in [1]. It was trained

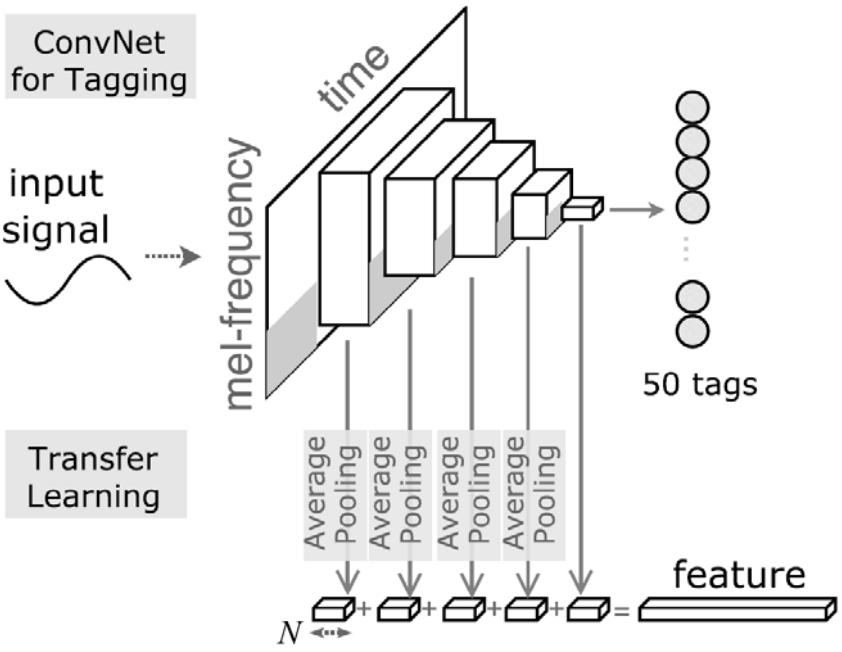


Figure 4.2: Overview of the feature extractor designed by Choi et al. [1]

on MSD, the largest available dataset in the MIR field. The trained model serves as a feature extractor. Figure 4.2 illustrates the feature extraction process. The pre-trained system takes mel-spectrogram representations of the audio signals as input. The model consists of five convolutional layers. After average pooling, every layer produces 32 features. Simply adding all 32 features per layer results in a one-dimensional 160-length feature vector. The feature vector thus consists of low-level as well as high-level features. Those features will be extracted for every of the DJ-Tags dataset's songs. It will then serve as the input of a newly designed multilayer-perceptron (MLP) serving as the classificator.

Afterwards, both approaches will be compared. On one hand, the labels of the DJCity datasets are heavily related to the labels of the target tasks and therefore a direct applicability on the target tasks is expected. On the other hand, the

DJCity datasets consist of just a fraction of training data, the auto-tagging task with MSD was trained on. The comparison of both approaches will give an insight for the universal suitability of the auto-tagging task on MSD as the source task in transfer learning setups.

4.3 Multitask Learning

Similarly to transfer learning, in a multitask learning setting the tasks take advantage of other tasks' knowledge. Other than the transfer learning approach, in multitask settings the tasks are trained simultaneously, learning a shared feature space. Dependent on the tasks' degree of similarity usually low-level feature representation are trained jointly. Embedding the shared space results in an either higher possible complexity of the features or smaller required training data. Several dissimilar multitask settings will be designed to evaluate the training process of the five tasks with the DJ-Tag dataset.

The following presented multitask settings correspond to either the CNN models for experiment 1 or the MLP that serves as the classifier in experiment 2. Figure 4.3 illustrates the tasks in a basic multitask setting. Training happens simultaneously, without sharing any layer's parameters. Multitask settings provide special characteristics. While the forward-passing takes place similar to singletask settings, the backpropagation process adjust the parameters in accordance to the loss of all tasks. This total loss is calculated as a sum of all losses. Another characteristic is the limited ability of adjusting hyperparameters for each individual task in the training process. Parameters like batch size or epochs are affecting all tasks.

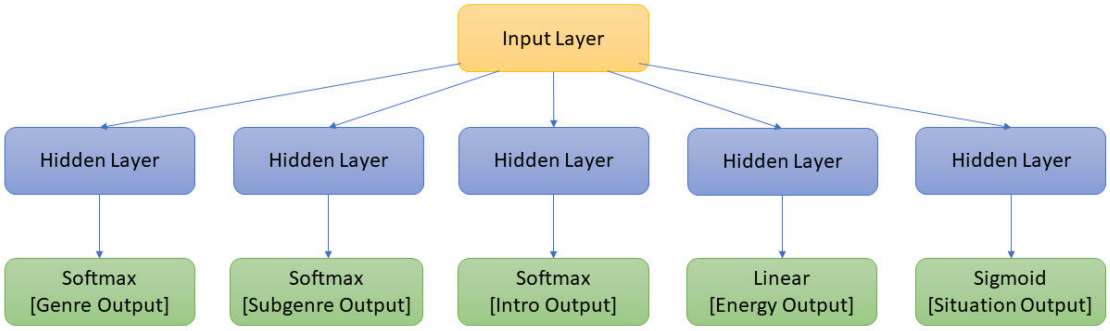


Figure 4.3: Proposed model design for multitask learning

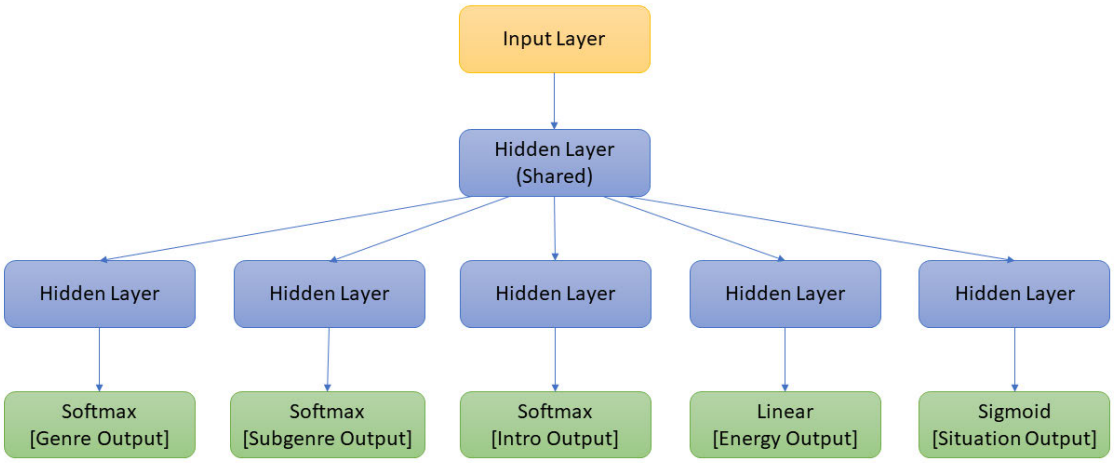


Figure 4.4: Proposed model for multitask learning with shared layers

In the second multitask setting the tasks share a low-level feature representation (figure 4.4). The number of shared layers can be altered. Since the tasks, despite main genre and subgenre, are not closely related, high-level features have to be designed individually to classify each task.

Another proposed multitask setting is supposed to take advantage of correlations and direct dependencies of the tasks. The subgenre labels indicate the main genre labels and vice versa. Also, some labels in the situation task (smallroom, mediumroom, bigroom) are only present for some of the main genres. To take the

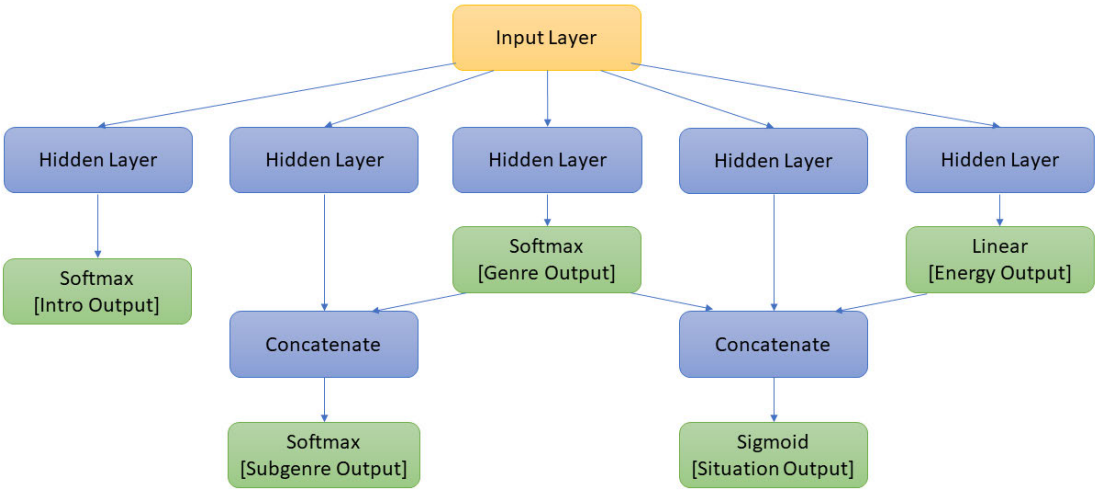
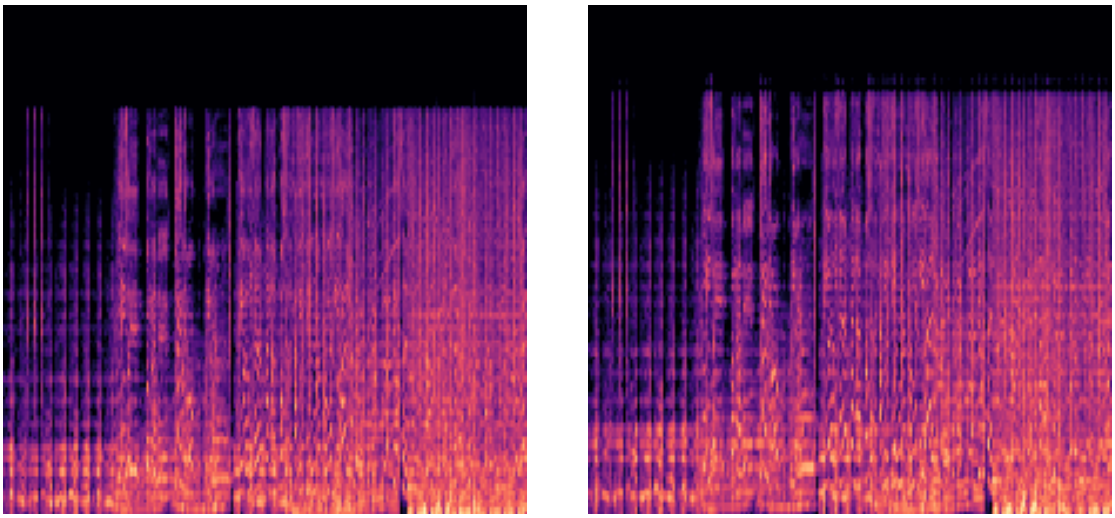


Figure 4.5: Proposed model for multitask learning with outputs of genre & energy branch as additional features for the subgenre & situation task

co-occurrences of some labels across the tasks into account, estimates of one task serve as an additional input for another task. In this example, the main goal is to help classifying the subgenre and situation tasks. Due to the large number of classes in the subgenre and the high imbalance in the situation task, both are expected to be difficult to train. Therefore, the main genre task is anticipated to help classifying both tasks. Additionally, due to an assumed correlation of energy and situation, the output value of energy is used as another additional feature for the situation task. Comparing the multitask variants among each other as well with singletask settings will give an insight of possibilities and problematic of multitask settings.

4.4 Data Augmentation

Apart from techniques of using the knowledge of other tasks and datasets to improve classification performances, one of the most common techniques for handling the difficulty of small datasets in deep learning setups is data augmentation. Data



(a) (b)

Figure 4.6: Mel-spectrogram of the song "Avicii ft. Aloe Blacc - S.O.S" (a) pitch shifted one tone down (b) pitch shifted one tone up

augmentation is a method used to increase the sample sizes. With several data processing techniques, the data samples are slightly modified in a way where the labels are still recognizable. For instance, in the image classification field, frequently used techniques for data augmentation are image transformations like rotating and cropping [25]. The objects in the images are still recognizable and the data, the network is trained on, can be increased. In the audio field data augmentation techniques exist as well. Popular data augmentation techniques include pitch shifting, time stretching or adding noise.

Aguirar et al. [26] explored these techniques to improve training CNNs for music genre classification. Compared to noise addition, time stretching and modified loudness, pitch shifting the audio samples one tone up and down proved to be most beneficial. Figure 4.6 visualizes the mel-spectrograms of a pitch-shifted example song. Following this outcome, augmenting data with pitch shifting one tone up

and down will be adopted in this work as well. With data augmentation applied on the DJ-Tag dataset, the sample size rises by a factor of three from 449 to 1347 samples. While comparing its influence on the improvement of the models' performances to the transfer learning and multitask learning approaches, the usage of data augmentation will then also serve to evaluate the implementation of the previously described approaches of transferring knowledge.

5 Experimental Setup & Results

5.1 Overview

The proposed models are categorized into two different approaches. For the first approach, CNNs were implemented. Training the CNN models on the DJ-Tags dataset without any of the proposed transfer and multitask learning methods serve as the baseline model of this work. This model was then pre-trained by the DJCity datasets corresponding to the first proposed transfer learning method. The second approach corresponds to using extracted features from a system trained on a large dataset. An MLP was implemented for classifying the tasks of the DJ-Tags dataset's tasks. The introduced multitask learning techniques are implemented in both approaches. Additionally, data augmentation was implemented in the proposed systems as well.

The models were trained on input data consisting of audio samples with a duration of 87 seconds. As part of the pre-processing , Choi et al. [3] trim the audio samples to 29 seconds. However, trimming the samples to 29 seconds is not expected to lead to satisfying results in the context of the presented dataset. The intro version task is characterized by the musical events at the beginning of the tracks. Classifying the other tasks, for example, the genre, based only on the beginning of the song is expected to be very difficult, especially if this song

Song extract (#features)	Total loss
0-29s (160)	6.28
29s-58s (160)	6.40
58s-87s (160)	6.48
0-87s (480)	6.03

Table 5.1: Total validation loss for the MTL model in regard of the song positions features are extracted from

excerpt consists only of acappellas or claps. On the other hand, extracting the samples of the middle of the tracks would result in an even harder challenge classifying the intro version. Due to the intro version task supposed to focus on the beginning of the tracks while the other tasks are supposed to rather focus on the data after the intro, songs were trimmed to threefold the length, resulting in 87 seconds long audio samples. Supporting the strategy, experiments on training the proposed MLP on only 29 seconds of the tracks led to worse results. Table 5.1 demonstrates the advantage of using 87 second against 29 seconds clips. The total losses correspond to the results of the later introduced MLP model in a multitask setting when training with different excerpts of the audio data. 87 second clips perform best overall. Focusing on the low frequencies of the data turns out to be more efficient [3]. Therefore, the songs were downsampled to 12 kHz to reduce the input dimension and the computational cost of the training process. The same procedure was performed with the augmented, pitch-shifted data.

Similar to the audio pre-processing, model design was highly inspired by the proposed systems for auto-tagging tasks in [3] as well. All models used the ADAM adaptive optimisation [27] and Rectified Linear Unit (ReLU) as the activation functions in the hidden layers. The output layers' activation functions were chosen according to the types of the classification tasks and range of the groundtruth

with the number of nodes according to the number of classes of the task. Softmax activation was implemented for the main genre, subgenre and intro-version classification as being singlelabel classification problems, sigmoid activation for situation classification as a multilabel classification problem and linear activation for the energy-level task being a regression problem [28]. Binary cross-entropy function was used for the multilabel classification task. For the multiclass problems, categorical cross-entropy and for the regression problem mean squared error was used as the loss function. In a multitask learning setup, the networks were trained on reducing the total loss of the tasks. This total loss was simply calculated as the sum of the validation loss of every individual task.

The total loss also serves as the metric to compare the overall performances of the proposed models. Even though the individual loss functions are different, they stay consistent in every model. Therefore comparing the networks performances with the total losses is valid with a smaller value indicating an overall better performing model. The evaluation metrics of the tasks depend on the classification types. Classification accuracy was used as the go-to evaluation metric for the multiclass tasks. In the next chapter, confusion matrices report better and worse performing individual classes. Table 5.2 gives insight on the values a random guess is expecting to score considering the knowledge of the class distributions. Similar to auto-tagging tasks [3] [17], Area Under the Receiver Operating Characteristic Curve (AUC) was used as the metric for the situation task as a multilabel classification task. AUC is robust to imbalanced data and works well when most classes are false in every sample. The range of AUC covers values from 0.5 to 1.0 with an AUC of 0.5 for a random guess and 1.0 for a perfect classification score. Additionally, discussing scores for precision, recall and F-measure give more insight into the performance of each individual class in the following chapter. For the regression

Task	Metric	Problem type	classes	Result
Main Genre	Multiclass	6	Acc	0.169
Subgenre	Multiclass	23	Acc	0.069
Intro	Multiclass	5	Acc	0.247
Energy	Regression	N/A	R^2	0
Situation	Multilabel	8	AUC	0.5

Table 5.2: Prior knowledge of the given tasks

task, R-squared (R^2) serves as the evaluation measure. R^2-values are expected to span between 0 and 1.0. Putting the value in the context of the other tasks' metrics, R^2 was prioritized over error measures like mean squared error due to 1.0 being the highest possible value, similarly to the previously described metrics.

Hyperparameter tuning is accomplished by trial and error. Training the designed CNN models in the first approach though was computationally demanding and time-consuming. Thus much more parameter tuning was possible to perform in the second experiment training the multilayer-perceptron. The parameters in the first approach are mainly based on the knowledge gained by the tuning parameters in the second experiment as well as on the proposed CNN models for auto-tagging by Choi et al. [3]. Performing a minimum of five test runs for every model, the optimal number of epochs for every test run was chosen before the validation loss did not decrease three times in a row. The models were then trained with the number of epochs corresponding to the mean of the optimal numbers of epochs during the test runs. Finally, at least five training runs in the first approach and ten in the second approach were performed. Evaluation was performed on the validation set. 80% of the dataset served as the training set and 20% for the validation. Training and validation test split was randomly created for every training run. The values for the evaluation metrics for every task, that are presented in the results, were

Layer	Output Shape	Parameters
Mel-spectrogram input	48 x 1874 x 1	0
3x3 Conv, 32 filters	48 x 1874 x 32	320
Max Pooling (2, 6)	24 x 312 x 32	0
Dropout (0.5)	24 x 312 x 32	0
Batch Normalization	24 x 312 x 32	128
3x3 Conv, 64 filters	42 x 312 x 64	18496
Max Pooling (4, 16)	6 x 19 x 64	0
Dropout (0.5)	6 x 19 x 64	0
Batch Normalization	6 x 19 x 64	256
3x3 Conv, 128 filters	6 x 19 x 128	73856
Max Pooling (6, 19)	1 x 1 x 128	0
Dropout (0.5)	1 x 1 x 128	0
Batch Normalization	1 x 1 x 128	512
Flatten	128	0
Dense (Softmax Activation)	6	774
Total Parameters		94,342

Table 5.3: Baseline CNN model for genre classification

calculated as the means of the evaluation metrics of the multiple training runs. All results including the applied hyperparameters, like learning rate, batch size or number of epochs are reported in this work's appendix.

5.2 Experiment 1: CNN with DJCity Datasets

5.2.1 Inputs & Architecture

In [3] Choi et al. evaluate several CNN architectures of different sizes for automatic tagging tasks. Although the models consisting of more convolutional layers perform better, the CNN in this work was inspired by the smallest proposed CNN model consisting of three convolutional layers. This was chosen due to the much smaller size of the dataset in this work. In table 5.3 the architecture of the designed CNN in a singletask setting is presented (genre classification in this example). The architectures for the other tasks vary only in the output layer's activation function, that was chosen as stated above. Facing the problem of heavy computational cost, reducing the models capacity was demanded and accomplished with reducing the number of filter maps of the CNN as shown in table 5.3 to 32, 64 and 128 for the first, second and third convolutional layer, respectively. Pilot experiments showed similar results to a network with higher numbers of feature maps while enormously decreasing computational cost. According to the model in [3] every convolutional block was extended with max pooling, decreasing the size of the feature maps to 1x1 and dropout of 0.5 with batch normalization, for regularization and preventing the model to overfit. Max-pooling has the ability to detect local features and was prioritized over avarage-pooling.

The network takes mel-spectrogram representations of the audio samples as input. They achieve better results than other representations, like short-time fourier transform (STFT) [3]. Being aligned to human auditory system, mel-spectrogram is suitable for subjective tasks and additionally more efficient in size. Further reducing the dimension of the input data was another strategy leading to a more efficient training process. Recent studies [29] evaluate the impact of frequency

and time resolution on accuracy and training times of models trained on auto-tagging tasks. Reducing the mel-bands to 48 and increasing hop size to 1024 samples results in a satisfying tradeoff between accuracy and computational cost and therefore was implemented in this work. The CNN presented in table 5.3 along with the CNNs for the other five tasks serve as the baseline models for this work.

This architecture remained consistent in the multitask learning settings. The first multitask learning model, henceforth simply termed as "multitask" or "MTL", consists of five branches of the same architecture as shown in table 5.3 with the outputs according to the individual task. Those branches belonging to every individual task share the same input. In the proposed multitask learning model with shared layers, henceforth termed as "shared layers", the branches share the first two convolutional blocks resulting in only the third convolutional layers' parameters as well as the output layers' parameters being individually trained for the specific task. The third proposed multitask learning method, named "Output-Feature", takes the estimation of the genre and energy tasks as additional features for the subgenre and situation tasks. More specifically, the values of the outputs of the genre branch after a training step serves as an additional feature for the input of the subgenre task's output layer and together with the energy branch output value for the input of the situation tasks output layer. It is important to point out that the gradient connecting the corresponding outputs has to be stopped during the backpropagation process to prevent updating the other task's (main genre and energy) weights.

For the transfer learning process the proposed baseline architecture was trained on the DJCity datasets as explained in the previous chapter. The first layers were

pre-trained with DJCity main genre data, shared by all five branches and the remaining layers of the main genre, subgenre and intro version tasks were pre-trained with the corresponding source datasets. First, second and third layer of the genre branches and second layers of the subgenre and intro version branches were set untrainable, while the other layers were then fine tuned by the target dataset with a ten times lower learning rate than training on the source task.

As the last proposed method to improve small datasets, data augmentation was implemented by training the baseline models with the additional created audio samples. For the final "combined" model, the introduced methods that increase the performance of classifying the tasks were brought together. The model consists of the first two convolutional layers being shared similarly to the shared model, while the weights of the layers were set correspondingly to the transfer learning process. Unlike the the previous transfer learning approach, all parameters were set trainable, increasing the performance. The input data was expanded by the samples created with data augmentation.

5.2.2 Results

Table 5.4 reports the results of the baseline singletask models. While the networks seemed to recognize some trend in the data for the genre, subgenre and situation tasks, the network performed poorly for the remaining tasks. Classification of the intro version was just a bit above random guess. No trend in the data for the energy was recognized. Increasing the sample size with data augmentation affected classifying the genres and energy marginally. However, slightly improving the situation task, the intro classification task benefited most with adding audio samples with increasing the accuracy up to 49%.

Task	Metric	Baseline	Data aug.
Main genre	Acc	0.38	0.38
Subgenre	Acc	0.11	0.13
Intro	Acc	0.27	0.49
Energy	R^2	-0.24	-0.06
Situation	AUC	0.62	0.69
Total Loss		6.79	6.08

Table 5.4: Results for the CNN singletask model without and with augmented dataset

Task	Metric	MTL	Output-Feature	Shared layers	Shared layers + data aug.
Main genre	Acc	0.34	0.37	0.36	0.42
Subgenre	Acc	0.11	0.11	0.12	0.17
Intro	Acc	0.29	0.27	0.29	0.53
Energy	R^2	-0.28	-0.47	0.06	0.16
Situation	AUC	0.65	0.64	0.65	0.75
Total Loss		6.81	6.78	6.65	5.91

Table 5.5: Results for the proposed mutltiask models

Training in a multitask setting results in similar performances. Using the estimates for main genre and energy as additional features for subgenre and situation did not improve performance. Due to the relatively poor performances of those tasks, especially the energy task, this comes not surprising. Note that the MTL and Output-Feature models are very similar and performances should only differ for the subgenre and situation tasks. The small differences in the values for the remaining tasks indicate an affiliation of the networks for the specific training and

Task	#classes	Metric	DJCity Dataset
Main genre	6	Acc	0.64
Subgenre	18	Acc	0.31
Intro	3	Acc	0.69

Table 5.6: Results for the source tasks

validation set that changes for every training run. The small number of training runs in this experiment might result in some tasks randomly over- or underperform based on the data the network is trained on and validated. The overall performance in the multitask setting improveed with sharing the first two convolutional layers across all tasks. Augmenting the audios in this setup resulted in the overall best performing model. Interestingly, this is the first model that seemed to slightly recognize a trend in the energy data.

For the transfer learning approach the models' parameters were pre-trained by the DJCity datasets. Results for the source tasks are reported in table 5.6. Thanks to the high number of samples, the model did a better job in predicting the main genres. Due to a smaller number of classes, direct comparison of the results for the source tasks and the target tasks are not possible for the remaining tasks. However, the prediction of intro version as well as subgenre benefited from the additional knowledge gained from the source tasks as reported in table 5.7. Even though the first layer was pre-trained by the main genre task, the situation classification tasks also seemed to benefit from the additional knowledge in the low-level feature representation. Surprisingly, the main genre task did not much improve despite correlating most with the related source task. Adding data augmentation, the performances of all tasks further improve. The combined model improves the performances of all tasks but the intro version task.

Task	Metric	TL DJCity	TL DJCity+data aug.	Combined
Genre	Acc	0.39	0.43	0.45
Subgenre	Acc	0.18	0.20	0.20
Intro	Acc	0.42	0.48	0.39
Energy	R^2	0.01	0.13	0.20
Situation	AUC	0.67	0.74	0.75
Total Loss		6.19	5.97	5.97

Table 5.7: Results for the transfer learning methods. *Combined* stands for the pre-trained model with shared layers and data augmentated input

In comparison to the results of the baseline models, data augmentation achieved the overall highest performance improvement in comparison to the other proposed methods. Without data augmentation, the transfer learning setting delivered better results than the presented multitask learning settings. Note that the transfer learning method is realised in a multitask setting as well. Then again, when training with more data, sharing layers achieved better results than the transfer learning method. Interestingly, the models did not recognize the main genres of the songs much better with the proposed methods, while slightly better results for the subgenres could be achieved. Intro classification benefits notably from the proposed methods, especially transfer learning and data augmentation. Same goes for the situation task. Most of the proposed models, however, fitted the data for energy very poorly, even worse than a random guess. Only the shared layers and the transfer learning models trained with augmented data seemed to recognize a trend in the data.

5.3 Experiment 2: MLP with MSD Features

5.3.1 Inputs & Architecture

In the second experiment, the transfer learning method extracting features from the pre-trained system on MSD was implemented. The system, originally trained on 29 seconds audios, takes mel-spectrogram representations with 96 mel-bins as input and extracts 160 features per sample [1]. Since 87 seconds clips were used as audio samples, 160 features were extracted three times for every 29 seconds of the audios resulting in a 480-feature vector for each song. Additionally, feature selection methods were implemented to reduce the dimensionality of input data. The motivation of using feature selection was to evaluate the influence of input dimensionality on training small datasets. A mix of feature selection algorithms, including removing quasi constant (variance threshold of 0.01) and highly correlated features (correlation threshold of 0.8), recursive feature elimination with cross-validated selection (RFECV), analysis of variance (ANOVA) f-value and mutual information (MI) was implemented to reduce the number of features. Evaluating combinations of the algorithms on the performances of each task was based on trial and error and results in applied algorithms as reported in table 5.8. With augmented data though, the resulting number of kept features differed.

A simple feed forward neural network with decreasing number of neurons per layer was implemented as a classifier. In comparison to [15] using one hidden layer for classifying genre in a similar transfer learning setup, two hidden layers were implemented in the proposed MLP. Extending the MLP by one layer enables the possibility of creating a jointly shared feature representation for the multi-task learning methods being introduced later on in this chapter. In comparison to smaller layer dimensions, pilot experiments revealed an appropriate number of

Task	Constant/ correlated	RFE-CV	MI	F-value	#features
Main genre	x	x	best 40 (40)	-	40 (40)
Subgenre	x	x	best 180 (60)	-	180 (60)
Intro	x	x	best 100 (70)	-	100 (70)
Energy	(x)	-	best 60% (30%)	-	288 (87)
Situation	x	x	-	best 16 for each label	90 (90)

Table 5.8: Feature selection algorithms used for the tasks. Values in parentheses correspond to multitask settings

Layer	Output Shape	Parameters
Input	480	0
Dense	64	30784
Dense	32	2080
Dense (Linear Activation)	1	33
Total Parameters		32,897

Table 5.9: MSD transfer learning model for energy classification

128 and 64 neurons for the first and second hidden layer, respectively. In the singletask learning setups, altering the number of neurons for the first and second layer improved performance on some tasks resulting in two different implemented architectures with 128 or 64 neurons for the first layer and 64 or 32 neurons for the second. Table 5.9 gives an overview for a singletask learning architecture (energy prediction in this example). With observing the behavior of validation and training loss, dropout and batch normalization layers were implemented in some configurations to counteract overfitting, similarly to the first experiment. Hyperparameters were adjusted on a trial and error method. Learning rates were chosen with observing the performances of learning rate values from 0.0001 to 0.1 in 6 steps. Learning rates of 0.005 and 0.001 performed best for the proposed models. In general, adding augmented data resulted in lower learning rates and higher number of epochs increasing the performances. The proposed network architectures with the corresponding hyperparameters and results are reported in this work's appendix.

Similarly to experiment 1, the architecture of the MTL configuration training all tasks simultaneously is based on the singletask architectures. Sharing the first hidden layer by the five multitask branches results in the "shared layers" configuration. Note that the shared layer's number of nodes has to be consistent for all branches. 128 nodes for the shared layer and 64 for the second, task specific layers were implemented. Pilot experiments revealed a better performance sharing the first of the two hidden layers than adding an additional shared layer to the network. Further multitask learning methods were evaluated. First approach was supposed to take advantage of the direct relationship between the main genre and the subgenre task with learning jointly two hidden layers. Just as presented in the first experiment, using the estimations of genre and energy as additional inputs for

the subgenre and situation tasks were implemented in the MLP as well. Additionally, the effect of pre-training the purposed networks with the DJCtiy Datasets on the model's performances is tested.

5.3.2 Results

The results for the proposed MLP networks with and without augmented data and feature selection algorithms are reported in table 5.10. Bold values indicate the best achieved results overall. In comparison to the networks in experiment 1 the proposed models achieved better overall performances and better or similar results for each task. Despite energy prediction in the MTL setting, feature selection improved performances for all tasks in every model. Increasing sample size with data augmentation vastly improved the performances thus clearly outperforming any models in experiment 1. Interestingly, while feature selection algorithms improved performances on the original dataset, with augmented data the networks performed better making use of all available features. Overall, training in a single-task setting led to better results than in multitask settings. In a multitask setting, sharing the first layer was the better option, while with data augmentation, despite for the main genre task, the MTL model performed better.

Additionally pre-training the singletask MLP networks with similar datasets did only improve the main genre task as reported in table 5.11. However, it became redundant when adding augmented data to the target task. In comparison to the first experiment, the source tasks perform better with the features extracted from the auto-tagging system trained on MSD.

Table 5.12 reports further ideas to improve performances in a multitask setting. Without data augmentation, the subgenre task took advantage of jointly train-

Task	Data Aug.	Feature Sel.	Metric	Singletask	MTL	Shared Layer
Main genre	no	no	Acc	0.56	0.41	0.42
	no	yes	Acc	0.63	0.48	0.50
	yes	no	Acc	0.77	0.76	**0.79**
	yes	yes	Acc	0.72	0.72	0.77
Subgenre	no	no	Acc	0.16	0.14	0.16
	no	yes	Acc	0.22	0.18	0.20
	yes	no	Acc	0.62	0.61	0.54
	yes	yes	Acc	0.60	0.59	0.53
Intro	no	no	Acc	0.50	0.49	0.42
	no	yes	Acc	0.55	0.53	0.52
	yes	no	Acc	**0.79**	**0.79**	0.76
	yes	yes	Acc	0.78	**0.79**	0.75
Energy	no	no	R^2	0.17	0.31	0.07
	no	yes	R^2	0.24	0.22	0.21
	yes	no	R^2	0.54	0.53	0.41
	yes	yes	R^2	**0.57**	0.48	0.43
Situation	no	no	AUC	0.74	0.71	0.73
	no	yes	AUC	0.78	0.72	0.75
	yes	no	AUC	0.87	0.87	0.84
	yes	yes	AUC	0.87	0.87	0.86
Total Loss	no	no		5.61	6.03	5.92
	no	yes		5.22	5.71	5.30
	yes	no		2.85	2.91	3.05
	yes	yes		3.09	3.17	3.14

Table 5.10: Results for the proposed MLP networks with the transfer learning system trained on MSD

Task	Source tasks	Pre-training 1st layer	Pre-training 1st & 2nd layer	Singletask
Main genre	0.70	0.59 (0.77)	0.57 (0.74)	0.56 (0.77)
Subgenre	0.39	0.16	0.15	0.16
Intro	0.71	0.46	0.42	0.50

Table 5.11: Results for additionally pre-training MLP models with the DJ City datasets in accuracy. Values in parentheses correspond to training with augmented input data

Task	Data aug.	Main & sub-genre	Output-feature	Singletask
Main genre (Acc.)	no	0.54		0.56
	yes	**0.81**		0.76
Subgenre (Acc.)	no	0.21	0.17	0.16
	yes	0.60	**0.63**	0.62
Situation (AUC)	no		0.76	0.74
	yes		**0.88**	0.87

Table 5.12: Results for further multitask methods to increase performance

Task	Metric	Singletask	MTL	Output-Feature
Main genre	Acc	**0.77**	0.76	**0.77**
Subgenre	Acc	0.62	0.61	**0.63**
Intro	Acc	**0.79**	**0.79**	**0.79**
Energy	R^2	**0.57**	0.53	0.53
Situation	AUC	0.87	0.87	**0.88**
Total Loss		2.85	2.91	**2.82**

Table 5.13: Overview of the best performing MLP networks taking augmented data without feature selection as input

ing with the main genre task sharing the first two layers. However, learning a shared space did not affect the main genre task positively. With a higher sample size though, the tasks behaved the other way round. Parallel to the shared layer model reported in table 5.10, the main genre task benefited from training a shared learning space. Considering that the primary idea of implementing the main genre task was to help the other tasks in a multitask setting, the focus here lay on the subgenre task and therefore this setting was not advantageous. Though only slightly, using estimates for the main genre and energy tasks improved the subgenre and situation tasks.

Complementing the MTL network with the Output-Features method resulted in the overall best MLP network with the lowest total loss value of 2.82 as shown in table 3.3. Note that the results for the tasks only differ marginally with highest accuracy values of 0.77, 0.63 and 0.79 for the main genre, subgenre and intro version, respectively and an AUC value of 0.88 for the situation task. Solely the energy task showed a considerably better performance in a singletask setting with a R^2 value of 0.57 towards a R^2 value 0.53 for the multitask networks.

6 Discussion

With implementing the proposed methods transfer learning, multitask learning and data augmentation, networks were able to improve the performances of classifying the tasks. Solely increasing the sample size by a factor of 3 with data augmentation techniques created a noticeable performance gain thus indicating the original dataset size to be too small to train the networks. Same goes for the transfer learning methods, leading to similar results, as reported in figure 6.1. Except for the subgenre task, transfer learning with a system trained on MSD as the source task achieved better results than pre-training the model with the DJCity datasets. Even the main genre and intro version tasks did not benefit as much from the highly correlated source tasks as with the MSD system. Thus clearly demonstrates the importance of the sample size of source tasks' datasets. Both transfer learning settings benefited from extra data. While training the pre-trained models with augmented data in experiment 1 slightly affects the classifications positively, additional data enormously increased performances in experiment 2 with the number of total loss fallen by half. It seems like a sweet spot of the required number of training data to achieve satisfying results was hit with this approach. Results from additionally pre-training the MLP with the DJCity datasets support the assumption. While increasing the performance of the similar main genre task trained with the original training set, pre-training did not affect the training with the augmented training set any more. In contrast, the additional data affecting

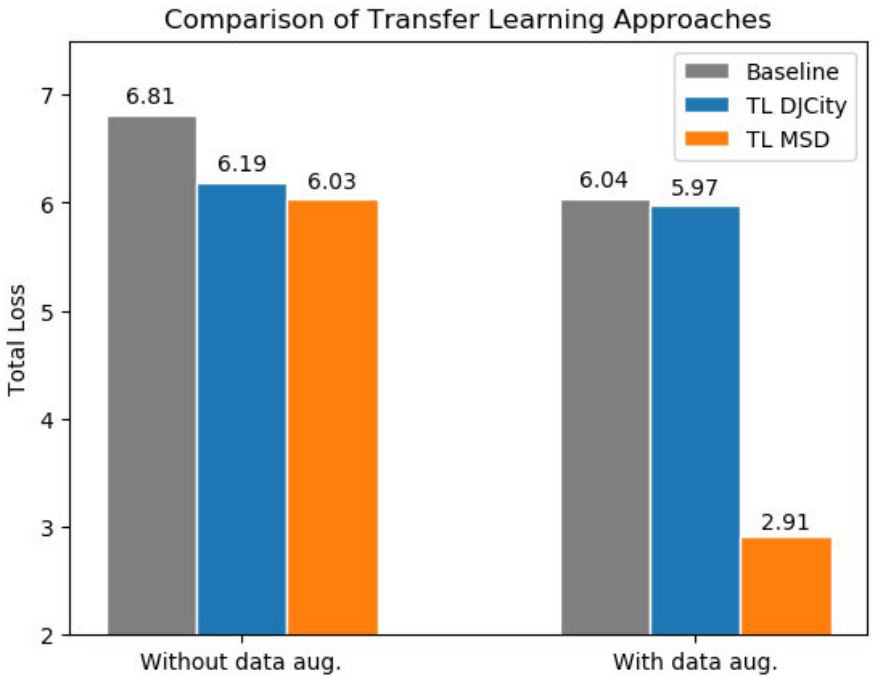


Figure 6.1: Comparison of the performances of the proposed transfer learning models in a basic multitask learning setting

the CNNs less, indicates that the number of training samples remained too small for training those kind of networks effectively. Or put the other way, the CNN models are still too complex for this amount of data.

While both data augmentation and transfer learning turned out to be advantageous throughout, the benefit of multitask learning settings on accurate classification was dependent on the input data, the interaction with transfer learning methods, and the explicit multitask learning variant. Simply training the tasks simultaneously did not improve the classification of the tasks, as reported in figure 6.2. Whereas training jointly low-level feature representations were able to reduce the complexity of the CNN thus improving classification on the original data samples. In experiment 2, sharing layers resulted in less accurate models,

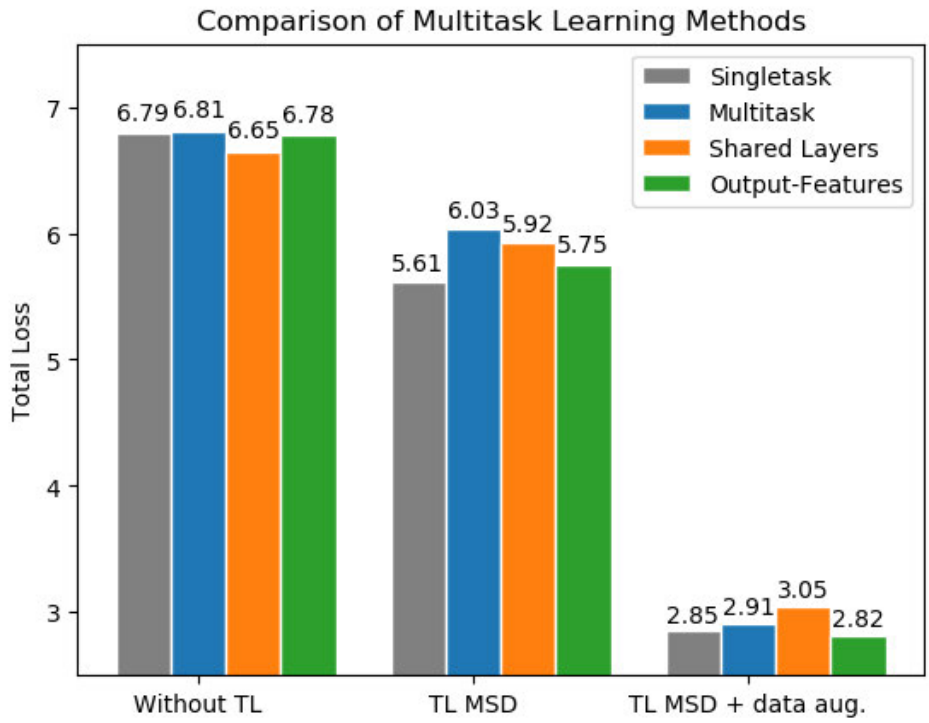


Figure 6.2: Comparison of the performances of the proposed multitask learning models

especially when trained with augmented data. The extracted features already include low-level knowledge gained on by far larger datasets. Therefore, the MLP serves only as a top level classifier being more effective to fine-tune the branches specifically for each individual task. Using estimates of one task as an additional feature for training another task positively effects training in a multitask setting throughout. One should not expect a performance gain though, when the tasks, supposed to help other tasks, are hardly recognized by the model, like in experiment 1. In experiment 2, with more accurate estimates for main genre and energy, the Output-Feature network performed best of all multitask learning methods and provided an even more accurate model than the singletask models. Multitask settings, in general were easier and faster to implement than singletask settings with less required test runs to find the optimal network architecture and hyperparam-

eters. On the other hand, layer dimensions in shared layers, hyperparameters like the number of epochs as well as the input data have to stay consistent in multitask settings. The possibility of singletask learning to design appropriate networks and hyperparameters more effectively for each individual task therefore led to more accurate models. Additionally, feature selection algorithms can be implemented specifically for each task when trained solely, turned out to be advantageous.

Experiment 2 shows, that reducing the input dimensions with feature selection is another beneficial method when training deep networks with only small available data. Feature selection algorithms were able to pick the most valuable features for each task thus facilitating the training process. For example, the intro version task is expected to focus on the beginning of the track and therefore features extracted for the first 29 seconds of the tracks are supposed to be more useful for this particular task. And indeed, it was observed, that 60% of the selected features by the recursive feature elimination algorithm were assigned to the first, 40% to the second and none of the features to the third 29 seconds of the tracks. On the other hand, with more training data, feature selection became redundant for most tasks with some tasks achieving better results making use of all available features.

The main difficulties experienced with training in multitask learning settings is the way, the training algorithms update the feature parameters of the network when the tasks are of a various nature and complexity. Different loss functions for multiclass, multilabel and regression problems had to be selected. The differences in the loss functions as well as in the number of classes of each task resulted in dissimilar magnitudes of the tasks' validation losses. Training the networks on reducing the total loss calculated as the sum of all individual validation losses led to focusing the training process on the tasks having the biggest part on the total

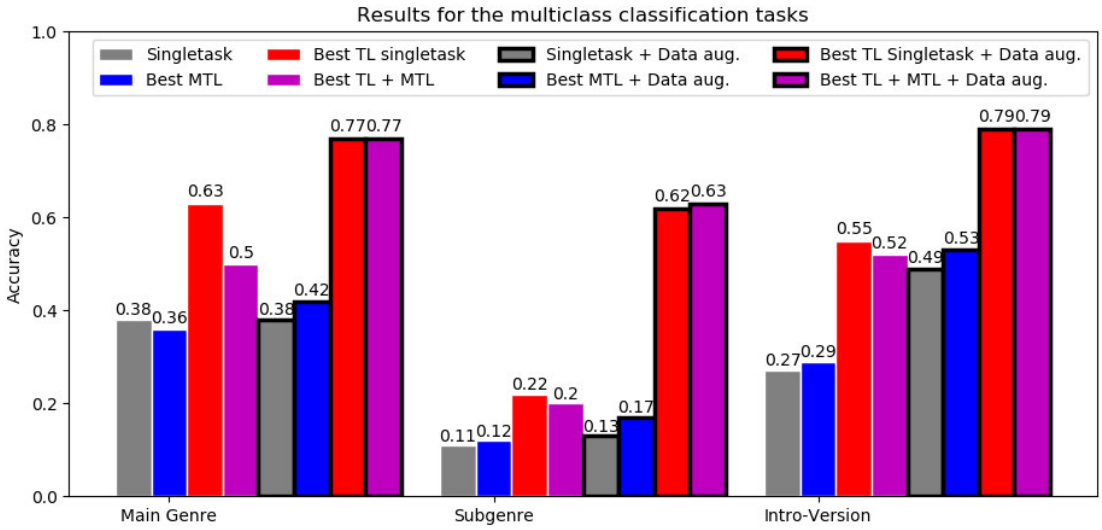


Figure 6.3: Accuracy scores of the main genre, subgenre & intro-version predictions for the proposed models

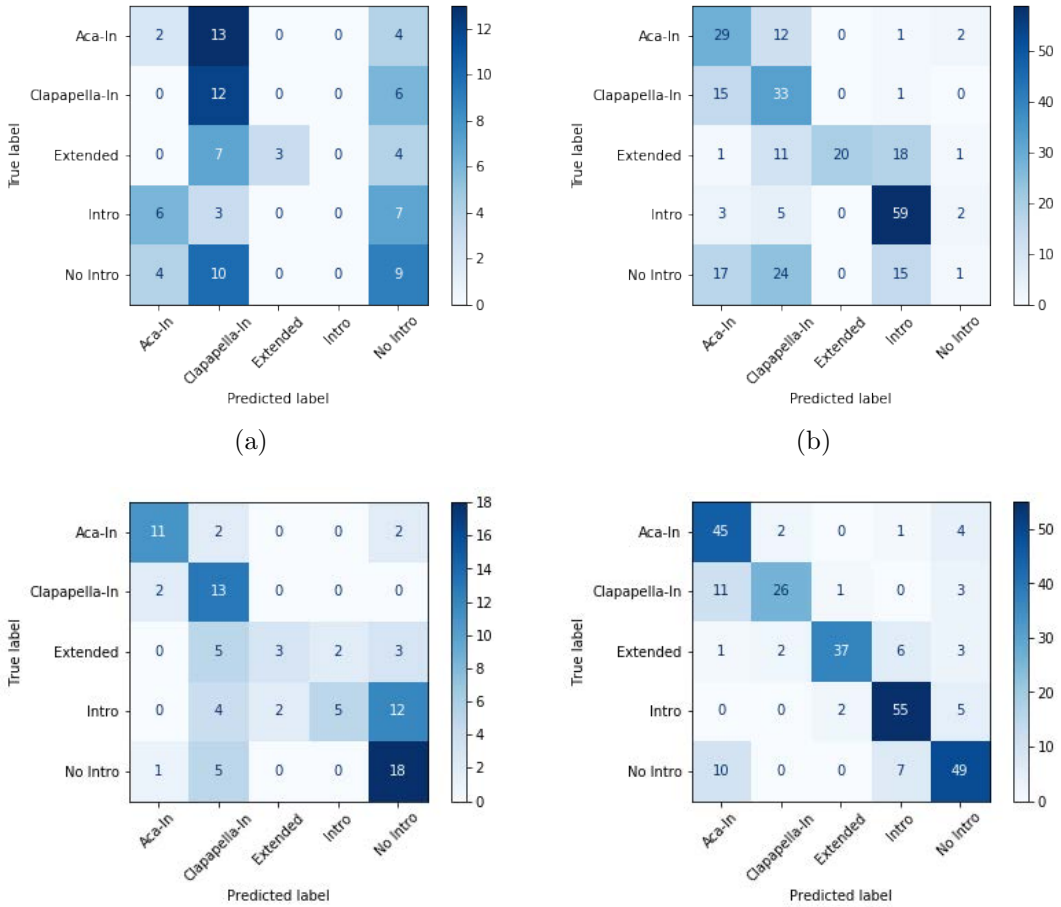
loss. Therefore, in the backpropagation process, weights were updated, also in task-specific layers, based on the performance of those tasks. The subgenre task, including 23 classes. throughout had the highest loss function thus dominating the whole training process. On the contrary, the mean squared error as the loss function for the energy prediction produced the smallest value. This circumstance affected the performances of each individual task and has to be considered when outlining the results.

Figure 6.3 illustrates the accuracy scores for main genre, subgenre and intro-version prediction. Transfer learning (red bars) and multitask learning approaches (blue) in most part led to more accurate models then singletask networks without transfer learning (grey) while transfer learning turned out to be superior to multitask. Training singletask networks with the extracted features from the auto-tagging task trained on MSD led to more accurate predictions than multitask networks (violet). Note that the right violet bar with black edges corresponds to the

Output-Feature model differing from the transfer learning singletask model mostly in implementing the estimates for main genre and energy as additional features for the subgenre and situation tasks. Therefore, this combination of transfer with multitask learning techniques affects especially these tasks. Adding augmented data (black edges) throughout increased performances.

Compared to the intro version task, higher accuracy scores were reached for main genres when trained with the original dataset size, although consisting of more classes. The difficulties of predicting intro versions may lie in the locally dependent nature of the determining musical events. Models predicting the main genres remained more accurate when trained on the extracted features from the auto-tagging task (red). This could be due to genre-related tags being most present in the Million Song Database. On the other hand, the intro version task profits enormously from the ability of a larger training set resulting in similar performances than the main genres. This leads to the assumption that the impact of the similarity between the source and the target task's data is higher with less available data, while with more data, training is easier for less complex tasks, as it is the intro version prediction in this study.

The confusion matrices in figure 6.4 give insight into the predictions of each individual intro version. As already mentioned, training on the original sample size turned out to be extremely difficult. The baseline model (CNN singeltask) was highly biased predicting mostly two classes (a). The CNN was not able to detect any useful data patterns. A lower bias can be observed with the transfer learning approach (c) but the results remained unsatisfying. With data augmented input and taking advantage of the knowledge from the other tasks with shared layers, the model recognized trends in the data (b). A bias was still present but



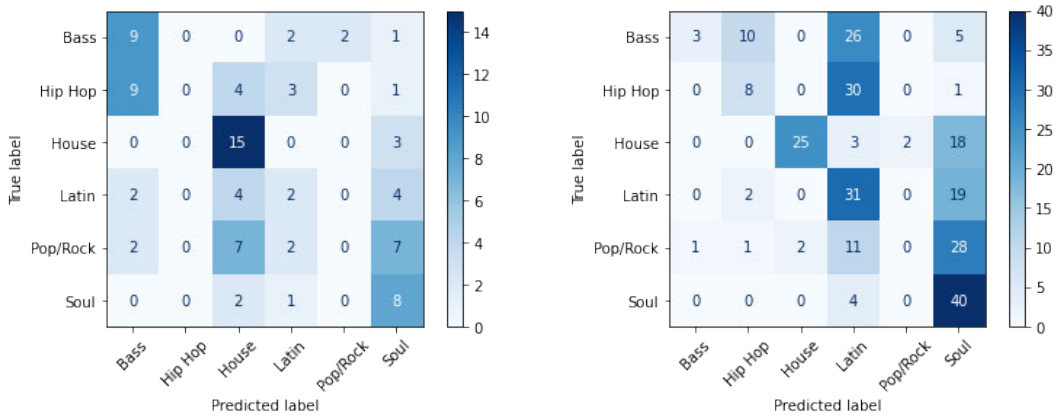
(a) (b)

(c) (d)

Figure 6.4: Confusion matrices for the intro version prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation

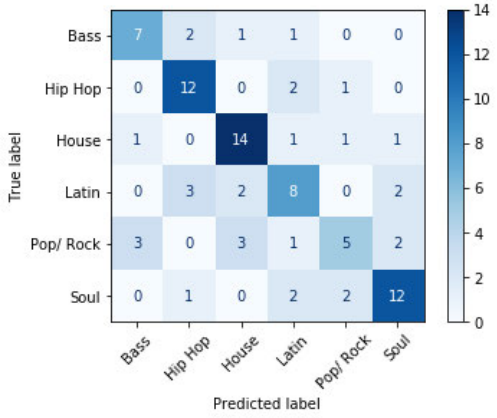
other than the No Intro class the intro versions in a large part were predicted accurately. In contrast, the transfer learning approach (d) clearly helped poorly recognized classes (No Intro) and was able to predict most of the classes accurately. Throughout all proposed models, correlations between Aca-In and Clapapella-In as well as Extended-In and Intro were observed. There indeed exist high similarities of those intro versions, making it difficult to distinguish them. Therefore, intro versions predicting the correlated class indicates, that the models were able to spot the appropriate musical data.

Equivalently to the intro versions, the models predicting the main genre were biased as well (figure 6.5) but in contrary, the transfer learning models (c, d) work pretty well even without additional data. Pop/ rock seems to be the most problematic main genre, performing worst throughout the models. This was expected, since pop and rock are highly dissimilar genres, therefore indicating that the models were able to recognize the appropriate feature representations. The main genres also consist of correlated genres across the classes, e.g. disco and disco house. Interestingly, the models were able to classify these genres into the right main genre categories. While other works were able to achieve accuracy scores above 90% for genre recognition with dataset sizes of a similar dimension as the proposed dataset [15], the complexity of the proposed dataset is extraordinary. A lot of DJ music and in particular the music in the proposed dataset correspond to song remixes, consisting of parts characterizing various genres. This makes the prediction of the genres exceptionally difficult. The results for training the DJCity Main Genre dataset highlight this problem. Though the dataset is relatively large in size, accuracy scores of only 64% for the CNN model and 70% for the MLP model could be achieved. Considering the complexity of the proposed datasets, accuracy scores above 70% can be considered as a satisfying result.

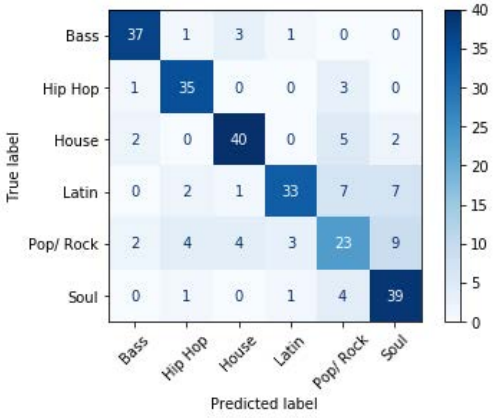


(a)

(b)



(c)

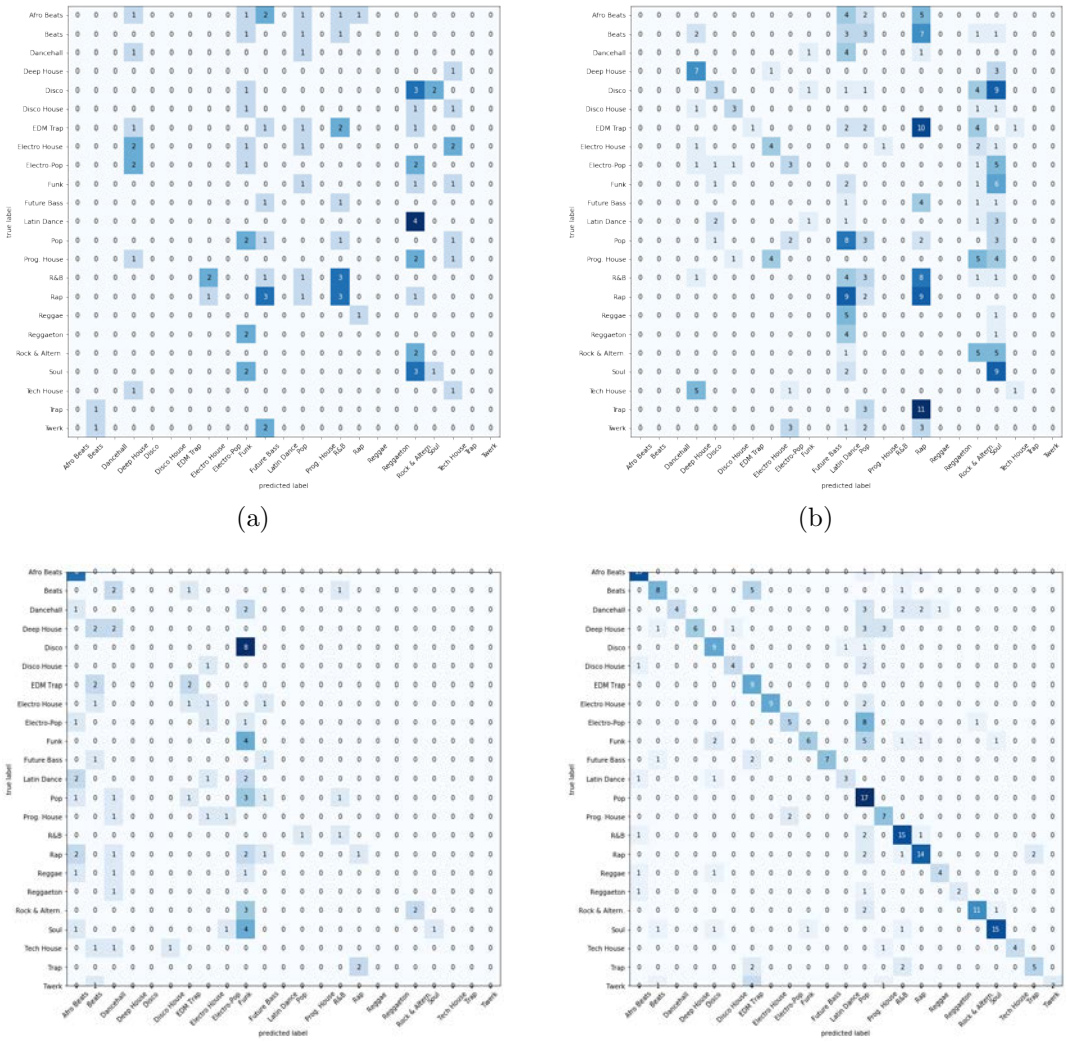


(d)

Figure 6.5: Confusion matrices for the main genre prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation

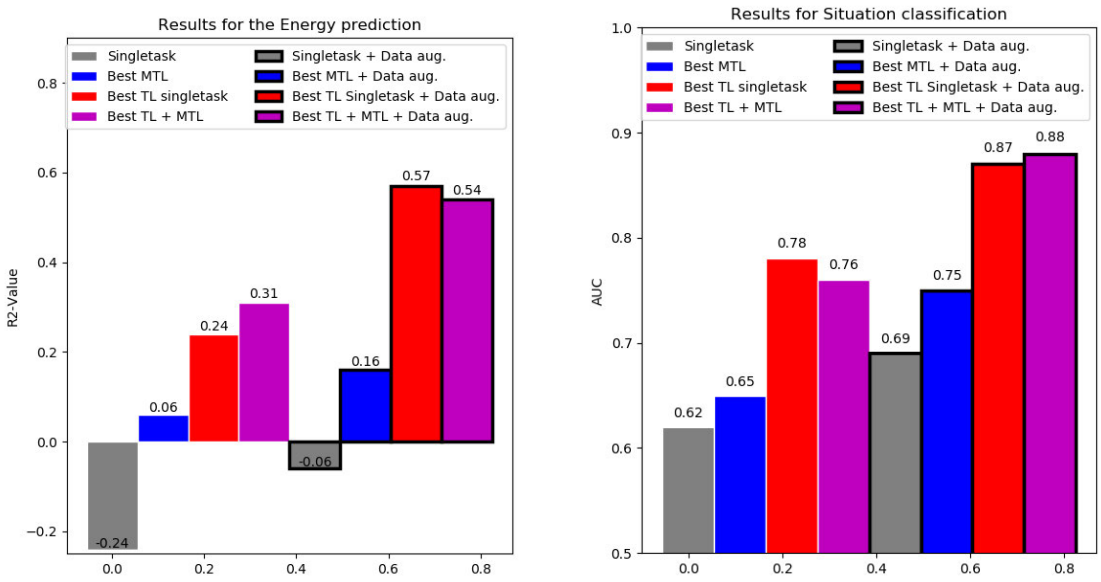
The subgenre task is especially interesting for this study. With a large number of classes (23) and an uneven distribution including subgenres with less than 10 samples, the networks were not expected to learn appropriate features. However, they seem to recognize a trend in the data already without the transfer learning approaches. With combining transfer learning with data augmentation, the performances increased drastically, as shown in figure 6.3. Taking into account the difficulties of this task, accuracy values above 60% are much higher than expected, leading to the assumption, that especially difficult tasks are able to benefit from the presented methods. Figure 6.6 demonstrates the performances of each individual subgenre. Due to the imbalance, the models mostly predicted subgenres with a high count, like pop, funk, soul or rap. Remaining some bias with a higher count of pop predictions, the best performing model (d) was pretty accurate in predicting each genre. The reasons for the relatively high performance of the subgenre tasks, considering the imbalance, the number of classes and a random guess of just 0.169 might be the close relationship to the tags in MSD as well as the prioritization over the other tasks in a multitask learning setting, as explained above.

The case is different for the energy task. The small impact on the training process in multitask settings reflected the results, as shown in figure 6.7 (a). In contrast to the other tasks, with augmented data and the transfer learning approach, energy prediction performed noticeably better in a singletask than in a multitask setting. Interestingly, without transfer learning, the energy task also benefited from a jointly training. Without the proposed methods, the prediction performed extremely poorly. The negative R^2 value points out, that the models did not recognize any trend in the data. But especially the data augmentation method in combination with the features extracted from the system trained on



(a) (b) (c) (d)

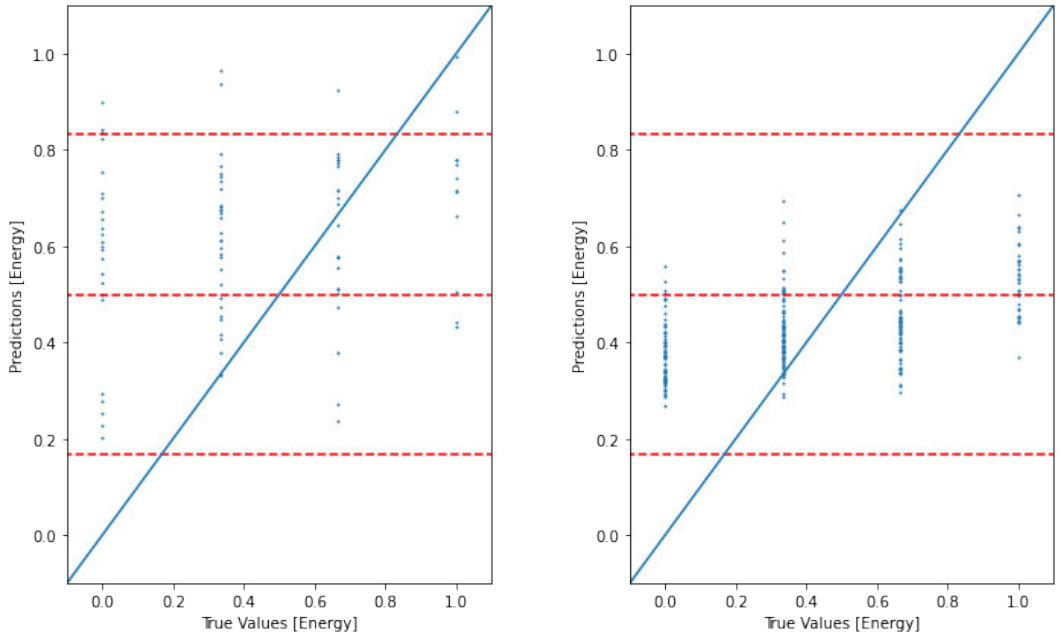
Figure 6.6: Confusion matrices for the subgenre prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selection, (d) MLP with MSD multitask Output-Feature model with data augmentation



(a) (b)

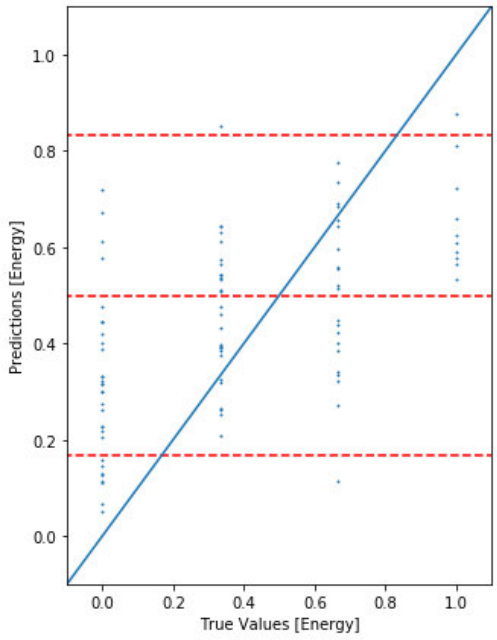
Figure 6.7: (a) R2 Scores of the energy prediction and (b) AUC scores of the situation prediction

MSD were able to achieve a performance gain. The R^2 for the best model is at a value of 0.57. In an automatic tagging system, where the energy would be tagged into the four presented energy levels, an accuracy score of 55% would be achieved for this model. The results for this task are therefore rather unsatisfying. One of the reasons could be the high subjectivity of this task. Scatter plots for various models predicting the energy are shown in figure 6.8. The red lines illustrate the border values for each of the four energy levels. While in the baseline model (a) the predictions were quite random, with adding data augmentation (b) the models learned to predict the energy values around the mean value of the training set. Some trend is recognizable though, giving the true values "0" lower prediction values as the true values "1". This trend is clearly visible with the transfer learning methods (c, d). However, the model remains to be prone to predict values around the mean of the training data.

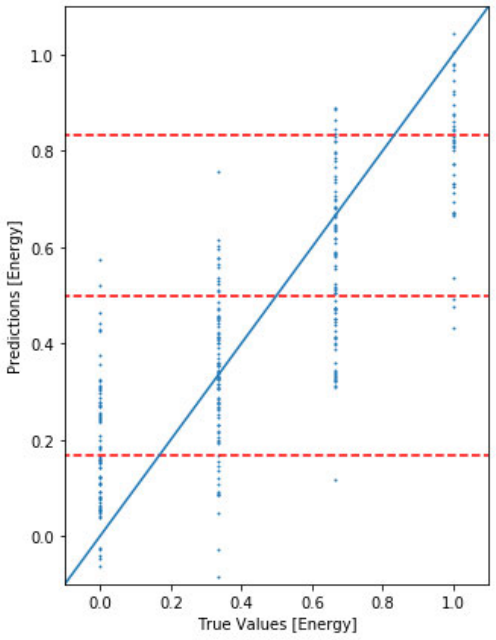


(a)

(b)



(c)



(d)

Figure 6.8: Scatter plots for the energy prediction task for the (a) CNN singletask, (b) CNN multitask model with shared layers and data augmentation, (c) MLP with MSD singletask model with feature selction, (d) MLP with MSD singletask model with data augmentation

Label	Precision	Recall	F1-Score	Support
Bigroom	0.00	0.00	0.00	7
Builddown	0.00	0.00	0.00	12
Lounge	0.00	0.00	0.00	19
Mediumroom	0.00	0.00	0.00	9
Midnight	0.42	0.85	0.56	34
Smallroom	0.50	0.08	0.14	12
Summer	0.14	0.10	0.12	20
Warmup	0.00	0.00	0.00	29

Table 6.1: Results for the situation prediction task with the baseline CNN model (AUC=0.62)

Label	Precision	Recall	F1-Score	Support
Bigroom	0.00	0.00	0.00	34
Builddown	0.00	0.00	0.00	44
Lounge	1.00	0.02	0.03	58
Mediumroom	0.00	0.00	0.00	28
Midnight	0.45	0.98	0.62	116
Smallroom	0.00	0.00	0.00	38
Summer	0.00	0.00	0.00	49
Warmup	0.00	0.00	0.00	59

Table 6.2: Results for the situation prediction task with the CNN multitask model with shared layers and data augmentation (AUC=0.75)

Predicting the appropriate DJ performance situation, being a multilabel problem with highly imbalanced data, was expected to be a particularly difficult task. Figure 6.7 though clearly illustrates, that with the three proposed methods, the performance, similar to the other tasks, could be increased. Taking into account the estimates for other tasks, predictions were more accurate. This demonstrates, that difficult multilabel classification problems can additionally learn from other tasks. More interesting though is the evaluation of the individual label perfor-

Label	Precision	Recall	F1-Score	Support
Bigroom	0.67	0.25	0.36	8
Builddown	1.00	0.27	0.43	11
Lounge	0.57	0.33	0.42	24
Mediumroom	0.00	0.00	0.00	12
Midnight	0.63	0.51	0.56	43
Smallroom	0.67	0.20	0.31	10
Summer	0.33	0.05	0.09	19
Warmup	0.50	0.29	0.36	14

Table 6.3: Results for the situation prediction task with the MLP with MSD singletask model with feature selection (AUC=0.77)

Label	Precision	Recall	F1-Score	Support
Bigroom	0.67	0.18	0.29	33
Builddown	0.93	0.31	0.47	45
Lounge	0.75	0.58	0.65	52
Mediumroom	1.00	0.03	0.06	31
Midnight	0.83	0.73	0.78	128
Smallroom	1.00	0.14	0.25	28
Summer	0.67	0.15	0.25	39
Warmup	0.86	0.24	0.37	51

Table 6.4: Results for the situation prediction task with the MLP with MSD multitask Output-Feature model with data augmentation (AUC=0.88)

mances. The models without transfer learning methods (table 6.1 & 6.2) were able to predict only a few of the labels. Due to the highest count, the models were prone to predict the label Midnight. The small precision value around 0.4 indicates though, that most of the predictions were wrong. This changes with training the models with the features extracted from the transfer learning system (table 6.3 & 6.4). Most of the predicted labels have precision values above 0.5. The small recall values for labels with fewer appearances reveal though, that also the best performing model rarely predicted those labels thus highlighting the difficulties of predicting multilabel problems with imbalanced data.

Model	TL/ MTL	Trainable param.	Training time	Feature extraction time
CNN MTL model	MTL	471,147	220ms/ step	4s/ track
CNN shared layer	MTL	395,115	61ms/ step	4s/ track
MLP MTL model	TL+MTL	165,739	11ms/ step	15s/ track

Table 6.5: Comparison of computational costs

Besides more accurate models, multitask and transfer learning systems provide the additional benefit of improving efficiency. Table 6.5 reports the savings of computational costs and time while training the models with the same equipment. Sharing low-level feature representations by all tasks resulted in a smaller number of trainable parameters and a more than three times faster training. Training the models with the extracted features from the transfer learning system led to even larger savings with around a third of the trainable parameters and 20 times faster training. On the other hand though, the feature extraction process lastet nearly four times longer than just extracting the mel-spectrogram for the CNN models.

Overall, the results in experiment 2 were more satisfying than in experiment 1. However, it has to be noted that due to the high computational costs of training CNNs, there were not many different architectures validated. On the other hand, a lot of model design and hyperparameter adjustments was accomplished in experiment 2. The complexity of the CNNs was much higher with around three times more trainable parameters than the MLP models. Reducing input dimensionality and network capacity led to higher performances in the second experiment, assuming that reducing the capacity of the CNNs in experiment 1 might increase performances as well. Therefore, the difference between the results both experiments provided could possibly be smaller with more time investment in designing the CNNs more appropriately.

7 Conclusion

This master's thesis evaluates the impact of transfer and multitask learning methods on music classification problems with small datasets. A new dataset consisting of various DJ-related tags was created. The tags were divided into several music label categories (genre, energy, situation, etc.) and classification problems (multiclass, multilabel, regression). Expecting, that label noise effects training on a much larger scale working with small datasets, a constrained vocabulary with expert annotations are highly recommended. To ensure the quality of the dataset, it was verified by an additional annotator.

A system trained on automatic song tagging with the Million Song Database served as the primary source task for transfer learning. Even though MSD includes mostly genre-related tags, the transfer learning approach also increased performances of less correlated tasks. Transfer learning with the system based on MSD clearly outperformed other transfer learning approaches presented in this work with more closely related source tasks but smaller amount of training data. Also, additional pre-training of the networks with closer related datasets was not advantageous, indicating the size of a source task dataset being of capital importance compared to semantic similarities. This work demonstrates the suitability of auto-tagging tasks on large dataset for transfer learning on music tagging tasks. These music classification tasks, similar to the tasks presented in this work, are mainly described on a high semantic level. Thus, the ability to adapt low- and

mid-level feature representations trained with MSD facilitates the training despite small amount of training data.

Easier tasks were able to assist more difficult tasks, like multilabel problems or tasks with imbalanced labels. Especially without the availability of transfer learning systems, multitask learning proved to be beneficial. Reducing the capacity of the networks resulted in more accurate models and efficient training. However, the networks did not benefit significantly when transfer learning and multitask learning approaches were combined. Anyway, they still provided very similar performances compared to singletask transfer learning settings. While an advantage of multitask settings on classification accuracies lay almost exclusively when trained without the transfer learning method, the use of both approaches can be suggested when training multiple and similar tasks. Multitask learning proved to be easier and faster to implement, additionally saving computational cost.

Increasing the sample size with data augmentation methods by a factor of three turned out to be beneficial as well. Moreover, training augmented data with the features extracted from transfer learning systems resulted in an enormous performance gain. It can be assumed, that with combining both approaches a close to perfect score for at least one task was achieved. Implementing more data augmentation techniques, again increasing the sample size, could further improve performances. I suggest to use this technique with caution though. Despite the signal processing techniques, with small counts for each labels, one can assume networks to memorize musical patterns associated to the specific sample rather than learning characteristics of the label category resulting in a suffered generalization. Transfer and multitask learning on the other hand might result in networks less prone to memorization.

One crucial problem identified but not treated in this work are the different loss functions that come with training various classification tasks in a simultane-

ously training setting. It turned out, that some of the tasks dominated the whole training process while other tasks having almost no impact on the training. In future works, I suggest to weight the validation losses and experiment with more hyperparameter tuning to improve the training on all tasks uniformly in multitask settings. Moreover, implementing neural networks with the ability of learning musical relevant relationships of labels across the label categories might help solve more complex music classification tasks.

Bibliography

[1] Choi, Keunwoo; György Fazekas; Mark Sandler; and Kyunghyun Cho (2017): "Transfer learning for music classification and regression tasks." In: *arXiv preprint arXiv:1703.09179*.

[2] Tzanetakis, George and Perry Cook (2002): "Musical genre classification of audio signals." In: *IEEE Transactions on speech and audio processing*, **10**(5), pp. 293–302.

[3] Choi, Keunwoo; George Fazekas; and Mark Sandler (2016): "Automatic tagging using deep convolutional neural networks." In: *arXiv preprint arXiv:1606.00298*.

[4] Lee, Jongpil; Jiyoung Park; Keunhyoung Luke Kim; and Juhan Nam (2017): "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms." In: *arXiv preprint arXiv:1703.01789*.

[5] Bertin-Mahieux, Thierry; Daniel PW Ellis; Brian Whitman; and Paul Lamere (2011): "The million song dataset." In: .

[6] Law, Edith; Kris West; Michael I Mandel; Mert Bay; and J Stephen Downie (2009): "Evaluation of algorithms using games: The case of music tagging." In: *ISMIR*. pp. 387–392.

[7] Schindler, Alexander and Peter Knees (2019): "Multi-Task Music Representation Learning from Multi-Label Embeddings." In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, pp. 1–6.

[8] Marchand, Ugo and Geoffroy Peeters (2016): "The extended ballroom dataset." In: .

[9] Çano, Erion and Maurizio Morisio (2017): "Music Mood Dataset Creation Based on Last. fm Tags." In: *2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria*.

[10] Schindler, Alexander; Rudolf Mayer; and Andreas Rauber (2012): "Facilitating Comprehensive Benchmarking Experiments on the Million Song Dataset." In: *ISMIR*. pp. 469–474.

[11] Pan, Sinno Jialin and Qiang Yang (2009): "A survey on transfer learning." In: *IEEE Transactions on knowledge and data engineering*, **22**(10), pp. 1345–1359.

[12] Argyriou, Andreas; Theodoros Evgeniou; and Massimiliano Pontil (2007): "Multi-task feature learning." In: *Advances in neural information processing systems*. pp. 41–48.

[13] Böck, Sebastian; Matthew EP Davies; and Peter Knees (2019): "MULTI-TASK LEARNING OF TEMPO AND BEAT: LEARNING ONE TO IMPROVE THE OTHER." In: *20th International Society for Music Information Retrieval Conference (ISMIR 2019)*.

[14] Weston, Jason; Samy Bengio; and Philippe Hamel (2011): "Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval." In: *Journal of New Music Research*, **40**(4), pp. 337–348.

[15] Ghosal, Deepanway and Maheshkumar H Kolekar (2018): "Music Genre Recognition Using Deep Neural Networks and Transfer Learning." In: *Interspeech*. pp. 2087–2091.

[16] Li, TL; Antoni B Chan; and Andy HW Chun (2010): "Automatic musical pattern feature extraction using convolutional neural network." In: *Genre*, **10**, p. 1x1.

[17] Kim, Taejun; Jongpil Lee; and Juhan Nam (2018): "Sample-level cnn architectures for music auto-tagging using raw waveforms." In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 366–370.

[18] Deng, Jia; et al. (2009): "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

[19] Sharif Razavian, Ali; Hossein Azizpour; Josephine Sullivan; and Stefan Carlsson (2014): "CNN features off-the-shelf: an astounding baseline for recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 806–813.

[20] Van Den Oord, Aäron; Sander Dieleman; and Benjamin Schrauwen (2014): "Transfer learning by supervised pre-training for audio-based music classification." In: *Conference of the International Society for Music Information Retrieval (ISMIR 2014)*.

[21] Lee, Jongpil and Juhan Nam (2017): "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging." In: *IEEE signal processing letters*, **24**(8), pp. 1208–1212.

[22] Zhang, Wenlu; et al. (2016): "Deep model based transfer and multi-task learning for biological image analysis." In: *IEEE transactions on Big Data*.

[23] Samala, Ravi K; et al. (2017): "Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms." In: *Physics in Medicine & Biology*, **62**(23), p. 8894.

[24] Choi, Keunwoo; György Fazekas; Mark Sandler; and Kyunghyun Cho (2017): "Convolutional recurrent neural networks for music classification." In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 2392–2396.

[25] Perez, Luis and Jason Wang (2017): "The effectiveness of data augmentation in image classification using deep learning." In: *arXiv preprint arXiv:1712.04621*.

[26] Aguiar, L Rafael; MG Yandre Costa; and N Carlos Silla (2018): "Exploring data augmentation to improve music genre classification with convnets." In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.

[27] Kingma, Diederik P and Jimmy Ba (2014): "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980*.

[28] Choi, Keunwoo; György Fazekas; Kyunghyun Cho; and Mark Sandler (2017): "A tutorial on deep learning for music information retrieval." In: *arXiv preprint arXiv:1709.04396*.

[29] Ferraro, Andres; Dmitry Bogdanov; Xavier Serra; Jay Ho Jeon; and Jason Yoon (2019): "How Low Can You Go? Reducing Frequency and Time Resolution in Current CNN Architectures for Music Auto-tagging." In: *arXiv preprint arXiv:1911.04824*.

Appendix

Results and the applied hyperparameters

For CNN Models (Experiment 1):

non-variable parameters:

Dropout: 0.5

Learning Rate: 0.005 (unless otherwise specified)

Batch size: 32

For MLP Networks with MSD-Features (Experiment 2):

2 Dropout & Batch Normalization layers for all multitask learning setting besides energy branch

2 Dropout & Batch Normalization layers for all singletask learning setting besides energy model & besides Genre/ Situation model with data augmentation

non-variable parameters:

Dropout: 0.5

Batch size: 32

Task	Metric	Result
Main Genre	Acc	0.38
Subgenre	Acc	0.11
Intro	Acc	0.27
Energy	R^2	-0.24
	MSE	0.139
Situation	AUC	0.62
Total Loss		6.80

CNN, Singletask models. Epochs: 10

Task	Metric	Result
Main Genre	Acc	0.38
Subgenre	Acc	0.13
Intro	Acc	0.49
Energy	R^2	-0.06
	MSE	0.118
Situation	AUC	0.69
Total Loss		6.08

CNN, Singletask models. Epochs: 10, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.34
Subgenre	Acc	0.11
Intro	Acc	0.29
Energy	Acc	0.29
	R^2	-0.28
	MSE	0.136
Situation	AUC	0.65
Total Loss		6.81

CNN, Multitask model, Epochs 15

Task	Metric	Result
Main Genre	Acc	0.40
Subgenre	Acc	0.13
Intro	Acc	0.54
Energy	Acc	0.33
	R^2	0.08
	MSE	0.103
Situation	AUC	0.72
Total Loss		6.04

CNN, Multitask model, Epochs 13, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.36
Subgenre	Acc	0.12
Intro	Acc	0.29
Energy	Acc	0.31
	R^2	0.06
	MSE	0.107
Situation	AUC	0.65
Total Loss		6.65

CNN, Shared layer Model, Epochs: 13

Task	Metric	Result
Main Genre	Acc	0.42
Subgenre	Acc	0.17
Intro	Acc	0.53
Energy	Acc	0.34
	R^2	0.16
	MSE	0.094
Situation	AUC	0.75
Total Loss		5.91

CNN, Shared layer Model, Epochs: 18, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.37
Subgenre	Acc	0.11
Intro	Acc	0.27
Energy	Acc	0.27
	R^2	-0.47
	MSE	0.152
Situation	AUC	0.64
Total Loss		6.78

CNN, Output-Feature model, Epochs: 15

Task	Metric	Result
Main Genre	Acc	0.42
Subgenre	Acc	0.12
Intro	Acc	0.36
Energy	Acc	0.31
	R^2	-0.06
	MSE	0.117
Situation	AUC	0.65
Total Loss		6.32

CNN, Multitask model, Pre-trained Genre Weights for layer 1, 2, 3, 4, Subgenre weights for layer 3, Intro weights for layer 3 (all not trainable), Epochs:8, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.44
Subgenre	Acc	0.14
Intro	Acc	0.39
Energy	Acc	0.30
	R^2	-0.05
	MSE	0.120
Situation	AUC	0.64
Total Loss		6.21

CNN, Multitask model, Pre-trained weights for: genre branch layer 1, 2, 3, subgenre branch layer 3, intro weights layer 3 (all not trainable), Epochs = 8, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.39
Subgenre	Acc	0.18
Intro	Acc	0.42
Energy	Acc	0.32
	R^2	0.01
	MSE	0.112
Situation	AUC	0.67
Total Loss		6.19

CNN, Multitask model, Pre-trained weights for: genre branch layer 1, 2, 3 (not trainable), subgenre branch layer 3 (trainable), intro weights layer 3 (trainable), Epochs = 9, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.43
Subgenre	Acc	0.20
Intro	Acc	0.48
Energy	Acc	0.35
	R^2	0.13
	MSE	0.094
Situation	AUC	0.74
Total Loss		5.97

CNN, Multitask model, Pre-trained weights for: genre branch layer 1, 2, 3 (not trainable), subgenre branch layer 3 (trainable), intro weights layer 3 (trainable), Epochs = 25, Learning Rate = 0.0005, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.40
Subgenre	Acc	0.17
Intro	Acc	0.39
Energy	Acc	0.30
	R^2	-0.06
	MSE	0.113
Situation	AUC	0.65
Total Loss		6.16

CNN, Shared layer model, Pre-trained weights for: genre branch layer 1 (shared, not trainable), 2 (shared, trainable), 3 (trainable), subgenre branch layer 3 (trainable), intro weights layer 3 (trainable), Epochs = 10, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.41
Subgenre	Acc	0.16
Intro	Acc	0.41
Energy	Acc	0.30
	R^2	-0.02
	MSE	0.117
Situation	AUC	0.67
Total Loss		6.29

CNN, Shared layer model, Pre-trained weights for: genre branch layer 1 (shared), 2 (shared), 3, subgenre branch layer 3, intro weights layer 3, (all trainable), Epochs = 10, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.40
Subgenre	Acc	0.13
Intro	Acc	0.38
Energy	Acc	0.24
	R^2	-0.277
	MSE	0.408
Situation	AUC	0.68
Total Loss		6.81

CNN, Shared layer model, Pre-trained weights for: genre branch layer 1 (shared), 2 (shared), 3, subgenre branch layer 3, intro weights layer 3, (all trainable), Epochs = 12, Learning Rate = 0.0005

Task	Metric	Result
Main Genre	Acc	0.44
Subgenre	Acc	0.17
Intro	Acc	0.46
Energy	Acc	0.35
	R^2	0.178
	MSE	0.089
Situation	AUC	0.75
Total Loss		6.03

CNN, Combined model, Pre-trained weights for: genre branch layer 1 (shared, not trainable), 2 (shared), 3, subgenre branch layer 3, intro weights layer 3, (all trainable), Epochs = 31, Learning Rate = 0.0005, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.45
Subgenre	Acc	0.20
Intro	Acc	0.39
Energy	Acc	0.33
	R^2	0.201
	MSE	0.089
Situation	AUC	0.75
Total Loss		5.97

CNN, Combined model, Pre-trained weights for: genre branch layer 1 (shared), 2 (shared), 3, subgenre branch layer 3, intro weights layer 3, (all trainable), Epochs = 31, Learning Rate = 0.0005, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.31
Subgenre	Acc	0.12
Intro	Acc	0.28
Energy	Acc	0.29
	R^2	-0.23
	MSE	0.139
Situation	AUC	0.61
Total Loss		6.80

CNN, Output-Feature model, Epochs = 13

Task	Metric	Result
Main Genre	Acc	0.34
Subgenre	Acc	0.12
Intro	Acc	0.23
Energy	Acc	0.31
	R^2	-0.22
	MSE	0.14
Situation	AUC	0.63
Total Loss		7,0

CNN, Output-Feature model, 2 layers shared, Epochs = 11

Task	Metric	Result
Main Genre	Acc	0.41
Subgenre	Acc	0.13
Intro	Acc	0.51
Energy	Acc	0.33
	R^2	0.14
	MSE	0.097
Situation	AUC	0.74
Total Loss		6,16

CNN, Output-Feature model, Epochs = 14, With data augmentation

Task	Metric	Result
Main Genre	Acc	0.39
Subgenre	Acc	0.14
Intro	Acc	0.53
Energy	Acc	0.32
	R^2	0.097
	MSE	0.098
Situation	AUC	0.73
Total Loss		6.06

CNN, Output-Feature model, 2 Shared layers, Epochs = 11, With data augmentation

Task	Arch.	FS	Data aug.	#ep	LR	Met	Rlt	Loss
Main Genre	64-32-6	no (480)	no	8	0.005	Acc	0.56	1.138
		no	yes	17	0.001	Acc	0.76	0.71
		yes (40)	no	8	0.005	Acc	0.63	1.025
		yes (208)	yes	10	0.001	Acc	0.71	0.87
Subgenre	128-64-23	no 480	no	28	0.001	Acc	0.16	2.78
		no	yes	20	0.001	Acc	0.62	1.223
		yes 180	no	28	0.001	Acc	0.22	2.64
		yes 180	yes	24	0.001	Acc	0.60	1.270
Intro	128-64-5	no (480)	no	12	0.005	Acc	0.50	1.164
		no	yes	14	0.001	Acc	0.79	0.55
		yes 100	no	10	0.005	Acc	0.55	1.08
		yes 100	yes	18	0.001	Acc	0.78	0.58
Energy	64-32-1	no (480)	no	9	0.005	R^2	0.17	0.092
						MSE	0.092	
						Acc	0.34	
		no	yes	20	0.001	R^2	0.54	0.052
						MSE	0.052	
		yes (288)	no	9	0.005	R^2	0.24	0.080
						MSE	0.0	
						Acc	0.38	
		yes (288)	yes	20	0.001	R^2	0.57	0.049
						MSE	0.049	
						Acc	0.55	
Situation	128-64-8	no 480	no	12	0.005	Acc	0.74	0.431
		no	yes	19	0.001	Acc	0.87	0.324
		yes (90)	no	12	0.005	Acc	0.78	0.396
		yes 88	yes	22	0.001	Acc	0.87	0.321

MLP Network with MSD-Features, Singletask

Task	Architecture	Metric	Result
Main Genre	64-32-6	Acc	0.41
Subgenre	128-64-23	Acc	0.14
Intro	128-64-5	Acc	0.49
Energy	64-32-1	R^2	0.31
		MSE	0.071
		Acc	0.44
Situation	128-64-8	AUC	0.71
Total Loss			6,03

MLP Network with MSD-Features, No Feature selection (480 Features), Epochs 11, Learning Rate = 0.005

Task	Architecture	Metric	Result
Main Genre	64-32-6	Acc	0.48
Subgenre	128-64-23	Acc	0.18
Intro	128-64-5	Acc	0.53
Energy	64-32-1	R^2	0.22
		MSE	0.080
		Acc	0.45
Situation	128-64-8	AUC	0.72
Total Loss			5.71

MLP Network with MSD-Features, With feature selection (191 features), Epochs = 13, Learning rate = 0.005

Task	Architecture	Metric	Result
Main Genre	64-32-6	Acc	0.75
Subgenre	128-64-23	Acc	0.61
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.53
		MSE	0.051
		Acc	0.54
Situation	128-64-8	AUC	0.87
Total Loss			2.91

MLP Network with MSD-Features, No feature selection (480 features), Epochs = 19, Learning rate = 0.001, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-64-5	Acc	0.79
Subgenre	128-64-23	Acc	0.61
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.55
		MSE	0.050
		Acc	0.54
Situation	128-64-8	AUC	0.88
Total Loss			2.83

MLP Network with MSD-Features, No feature selection (480 features), Epochs = 19, Learning rate = 0.001, With data augmentation

Task	Architecture	Metric	Result
Main Genre	64-32-6	Acc	0.71
Subgenre	128-64-23	Acc	0.59
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.48
		MSE	0.056
		Acc	0.53
Situation	128-64-8	AUC	0.87
Total Loss			3.17

MLP Network with MSD-Features, With feature selection (155 features), Epochs = 20, Learning rate = 0.001, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.42
Subgenre	128-64-23	Acc	0.16
Intro	128-64-5	Acc	0.42
Energy	128-64-1	R^2	0.07
		MSE	0.107
		Acc	0.37
Situation	128-64-8	AUC	0.73
Total Loss			5.92

MLP Network with MSD-Features, Shared layer, No feature selection (480 features), Epochs = 14, Learning Rate = 0.005, No data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.50
Subgenre	128-64-23	Acc	0.20
Intro	128-64-5	Acc	0.52
Energy	128-64-1	R^2	0.21
		MSE	0.086
		Acc	0.36
Situation	128-64-8	AUC	0.75
Total Loss			5.30

MLP Network with MSD-Features, Shared layer, With feature selection (191), Epochs = 14, Learning Rate = 0.005, No data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.79
Subgenre	128-64-23	Acc	0.54
Intro	128-64-5	Acc	0.76
Energy	128-64-1	R^2	0.41
		MSE	0.066
		Acc	0.42
Situation	128-64-8	AUC	0.84
Total Loss			3.05

MLP Network with MSD-Features, Shared layer, No feature selection (480), Learning Rate = 0.001, Epochs = 20, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.77
Subgenre	128-64-23	Acc	0.53
Intro	128-64-5	Acc	0.75
Energy	128-64-1	R^2	0.43
		MSE	0.061
		Acc	0.43
Situation	128-64-8	AUC	0.86
Total Loss			3.14

MLP Network with MSD-Features, Shared layer, With feature selection (155), Learning Rate = 0.001, Epochs = 28, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.77
Subgenre	128-64-23	Acc	0.63
Intro	128-64-5	Acc	0.80
Energy	64-32-1	R^2	0.54
		MSE	0.052
		Acc	0.55
Situation	128-64-8	AUC	0.88
Total Loss			2.82

MLP Network with MSD-Features, Output-Features, No feature selection (480 features), Epochs = 15, Learning rate = 0.001, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.74
Subgenre	128-64-23	Acc	0.63
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.47
		MSE	0.057
		Acc	0.52
Situation	128-64-8	AUC	0.88
Total Loss			3.08

MLP Network with MSD-Features, Output-Features, With feature selection (155 features), Epochs = 21, Learning rate = 0.001, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.81
Subgenre	128-64-23	Acc	0.58
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.53
		MSE	0.051
		Acc	0.54
Situation	128-64-8	AUC	0.88
Total Loss			2.84

MLP Network with MSD-Features, Main Genre Subgenre Shared + Output-Feature No feature selection (480), Learning Rate = 0.001, Epochs = 18, With data augmentation

Task	Architecture	Metric	Result
Main Genre	128-32-6	Acc	0.78
Subgenre	128-64-23	Acc	0.53
Intro	128-64-5	Acc	0.79
Energy	64-32-1	R^2	0.51
		MSE	0.055
		Acc	0.52
Situation	128-64-8	AUC	0.86
Total Loss			3.03

MLP Network with MSD-Features, Main Genre Subgenre Shared, No feature selection (480), Learning Rate = 0.001, Epochs = 17, With data augmentation

Hiermit versichere ich an Eides statt, dass ich die vorliegende Masterarbeit ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht. Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Berlin, den .....

. . . . . .

(Unterschrift)