ON THE IMPACT AND INTERPLAY OF INPUT REPRESENTATIONS AND NETWORK ARCHITECTURES FOR AUTOMATIC MUSIC TAGGING

Maximilian Damböck
Faculty of Informatics
TU Wien, Austria
maxi.damboeck@gmail.com

Richard Vogl
Faculty of Informatics
TU Wien, Austria
richard.vogl@tuwien.ac.at

Peter Knees
Faculty of Informatics, TU Wien, AT
School of Music, Georgia Tech, USA
peter.knees@tuwien.ac.at

ABSTRACT

Automatic music tagging systems have once more gained relevance over the last years, not least through their use in applications such as music recommender systems. State-of-the-art systems are based on a variant of convolutional neural networks (CNNs) and use some type of time-frequency audio representation as input, in a fitting combination, to predict semantic tags available through expert or crowd-based annotation. In this work we systematically compare five widely used audio input representations (STFT, CQT, Mel spectrograms, MFCCs, and raw audio waveform) using five established convolutional neural network architectures (musicnn, VGG-16, ResNet, a squeeze and excitation network (SeNet), as well as a newly proposed musicnn variant using dilated convolutions) for the task of music tag prediction. Performance of all factor combinations are measured on two distinct tagging datasets, namely MagnaTagATune and MTG Jamendo. A two-way ANOVA shows that both input representation and model architecture significantly impact the classification results. Despite differently sized input representations and practical impact on model training, we find that using STFT as input representations provides the best results overall. Furthermore, the proposed dilated convolutional architecture shows significant performance improvements for all input representations except raw waveform.

1. INTRODUCTION

Music tagging is a multi-label classification task to predict semantic high-level tags like “rock”, “piano” or “fast” for music pieces. As for many music information retrieval (MIR) tasks, deep learning has become the state-of-the-art for music tagging [1]. Specifically, CNNs have provided the best results for this task, using either types of audio spectrograms or raw waveform as network input [2, 3].

The selection of audio input representations for neural networks is often decided heuristically or by falling back to standard spectrograms and usually not evaluated [4]. However, the choice of input representation can have a significant impact on performance, as shown in other tasks like natural language processing (NLP), justifying also an evaluation of different representations [5].

While existing work compares different aspects of neural network architectures and input representations (see section 2), an evaluation of their interplay is missing. In this work, as our first contribution, we systematically compare five different architectures and five different input representations in a full-factorial design. The baseline architectures are: *(i)* VGG-16 [6], *(ii)* ResNet [7], *(iii)* SENet [8], *(iv)* musicnn [9], and *(v)* a musicnn architecture with dilated convolutional layers. Five different input representations are tested on each of those architectures: *(i)* raw waveform, *(ii)* short time Fourier transformation (STFT), *(iii)* Mel spectrogram, *(iv)* Mel frequency cepstral coefficients (MFCCs), and *(v)* constant-Q transformation (CQT) [2, 3, 10]. All experiments are repeated on two different datasets and a statistical analysis is used to test if differences of results are significant.

Through this extensive evaluation, we can state that our second contribution is the introduction of dilated convolutional layers in a music tagging network architecture (musicnn). This observed improvement can be related to the fact that the dilated layers allow for a large receptive field when only using a limited amount of layers, cf. [10].

Next, in section 2, we review related work, followed by an outline of the five input representations, five neural network architectures, and datasets used in section 3. Section 4 describes the experimental setup, training, and evaluation strategy and presents the statistical evaluation and significance analysis of the results. Finally, conclusions on the results and findings are drawn in section 5.

2. RELATED WORK

In [2], the authors evaluate state-of-the-art CNN models for music auto-tagging either by using the audio waveform or Mel spectrogram as input representations. MagnaTagATune (MTAT), Million Song dataset (MSD), and MTG-Jamendo (MTGJ) serve as datasets for training and evaluation. Among others, the study considers the following model architectures: A fully convolutional network, *musicnn* (cf. section 3.2.4), a Sample-Level CNN to process raw waveforms using squeeze- and excitation blocks (cf. section 3.2.3), a convolutional recurrent neural network, as well as a self-attention based approach using an adaptation of the transformer architecture. Furthermore, the work fo-

Attribution: M. Damböck, R. Vogl, and P. Knees, “On the Impact and Interplay of Input Representations and Network Architectures for Automatic Music Tagging”, in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

cuses on effects of data augmentation like pitch shifting or time stretching to improve generalization abilities. Note that while we follow a similar strategy wrt. evaluated architectures in our work, we focus on the impact of different input representations on these different architectures and evaluate their interplay.

More recently, transformer networks have been applied for music tagging [11,12]. While these models improve the state-of-the-art performance, they are significantly more complex than plain convolutional models. Therefore, we consider them not a good match for the evaluation in this work. Other deep learning techniques like transfer-learning, data augmentation, and model aggregation which have been used in music tagging [13], are mainly model agnostic and can be applied on top of any input representation and model combination.

In [14], the authors tested the effect of training data size for music tagging. They trained models with dataset sizes ranging from 100k to 1.2 million training instances using the private 1.2M-Songs dataset. Two CNN architectures are used, one for raw waveforms and a second one for Mel spectrograms inputs. The Mel spectrogram-based model outperformed the raw waveform-based model in both datasets on all tested configurations, which encouraged the conclusion that domain knowledge can be beneficial for designing model architectures.

Using dilated convolutions has shown promising results in image segmentation- and classification over the last few years [15–17]. A similar trend can be observed for the audio domain [18–21]. Dilation can be used to increase the receptive field using only a few layers, which can be beneficial for tasks focusing on global properties of the input.

In [22], the authors use dilated convolutions for environmental sound classification. This task (similarly to music tagging) requires an architecture with a large receptive field. Previously this was achieved by using very deep CNNs. The evaluation shows performance improvements of up to 10% on different datasets compared to existing deep CNNs while reducing model size.

In [4], the authors compare the effects of several audio preprocessing methods for music auto-tagging when using a CNN. For this, the same neural network was trained on the MSD using either STFT or Mel spectrogram as input representations. The evaluation shows no significant differences in performance for the two input representations. However, using log-compressed magnitude spectrograms showed a significant performance increase for all tested configurations.

A more comprehensive comparison of spectrogram representations for audio was conducted in [10] in the context of environmental sound classification. Different configurations for STFT, Mel spectrogram, CQT, continuous wavelet transform, and MFCCs were evaluated in the experiments, using different CNN configurations for training. The evaluation shows that in nearly all configurations, shallower CNN models outperformed deeper ones. The authors suspect that overfitting of the deeper model is the cause of this. The results also show that Mel spectrograms performed consistently well, while STFT and CQT did not, while MFCCs performed worst.

The effects of the reduction of frequency and time resolution of Mel spectrograms for CNN architectures in the context of music auto-tagging were investigated in [23]. Two different state-of-the-art CNN architectures served as a reference for the comparison of Mel spectrograms with different frequency and time resolutions. The results show that while reducing the frequency bands from 128 to 48, the performance loss was negligible.

3. METHOD

This section covers the description of the input representations as well as the neural network architectures used in the evaluation. The experiments are implemented in Python, using librosa [1] for audio processing and TensorFlow as deep learning framework. The source code is available online, refer to the source code for all implementational details [2] , there are also additional figures on the accompanying website [3] .

3.1 Input Representations

In the following section, the used input representations are discussed and parameter settings are provided. In this work, standard parameter settings established in the literature are used. Figure 1 visualizes the 2D input representations using a 10s audio snippet.

3.1.1 Raw Waveform

In the context of this work, the audio material is resampled to 16kHz mono. This is used consistently also for the source of all spectrograms. As waveform input representation, 16kHz mono 32bit float pulse-code modulation (PCM) data is used.

3.1.2 STFT

A STFT is calculated by repeatedly applying a discrete Fourier transformation (DFT) to small frames of the audio signal. As a baseline spectrogram, in this work, the librosa STFT with default parameters is used: frame size of 2 048 at 16kHz, a hop length of 512, and a Hann window function.

3.1.3 Mel Spectrogram

A usual modification to plain spectrograms is to use psychoacoustically-motivated frequency scales. A common choice is the Mel scale, which has a logarithmic characteristic. In this work, 96-frequency-bin Mel spectrograms are used. A minimum frequency of 32Hz and 12 bins per octave are used. These or similar values are consistently used throughout the literature [23].

3.1.4 MFCC

MFCCs are a compact representation reducing the frequency dimension of the Mel spectrogram using the discrete cosine transform (DCT). MFCCs have been successfully applied in speech recognition and to some extend

[1] https://librosa.org

[2] https://gitlab.com/MaxDamb/dl-autotagging

[3] https://tinyurl.com/ismir22-317

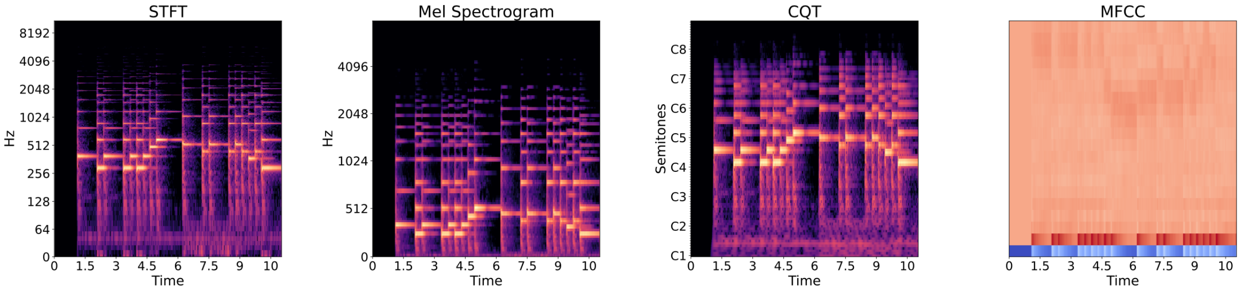

Figure 1. Different 2D input representations, from left to right: STFT, Mel spectrogram, CQT, and MFCC

also in MIR. The resulting coefficients describe the overall shape of the spectrogram columns [24]. Often the first 20 coefficients are used, which is also the default value in librosa, and therefore used in this work.

3.1.5 Constant-Q Transformation

Similar to the DFT, the CQT can be used to extract a time-frequency representation. The CQT results in a log-frequency scale and needs no further frequency transformation to match human audio perception. It is calculated similar to the STFT but enforces a constant ratio of the center frequency to resolution (constant-q) [25]. All parameters are chosen equal to comparable parameters of the Mel spectrogram: hop size 512, minimum frequency 32Hz, 12 bins per octave, and 96 frequency bins,

3.2 Network Architectures

In this section the five CNN architectures used for the experiments are discussed. Whenever kernel sizes are discussed, e.g. 3x3 filter kernels, these refer to two-dimensional inputs and imply 3(x1) filters for one-dimensional inputs. For more implementational details please refer to the original works, or the source code of the implementations.

3.2.1 VGG-16

The VGG-16 architecture introduced in [6] is used as a CNN baseline in this work. It consists of five convolutional blocks, each with 2-3 convolutional layers with 3x3 kernels and max-pooling with 2x2 kernels, followed by fully connected (FC) output layers. For each block the number of channels increases: 64, 128, 256, 512, and 512. For the FC layers (4 096 neurons each) the flattened output of the last convolutional block is used as input. Finally 50 sigmoid neurons form the output layer responsible for predicting the tags [6]. ReLU is used as an activation function for all hidden layers, and dropout- and batch-normalization layers are added after each block for regularization.

3.2.2 ResNet

Generally speaking, deeper networks have more processing power, but at the same time are harder to train because of the vanishing gradient problem. Residual networks implement so-called skip connections which are a solution to this problem and allow to train deeper networks [7]. A skip connection is a shortcut that adds the input to the output of a convolutional block or layer.

In this work, the ResNet-101 architecture (101 convolutional layers) [7] is used and reimplemented in TensorFlow. The basic building block is a bottleneck which consists of three convolutional layers (1x1, 3x3, 1x1) and a skip connection from the input to the output of the block. The 1x1 layers reduce the input dimensions by a quarter for the 3x3-layer and increase it again afterwards, which is done to reduces the memory footprint of the network. After the first convolutional layer (7x7) to downsize the input, the architecture consists of 3 blocks with 64 channels, 4 with 128, 23 with 256, and 3 blocks with 512 channels, followed by the sigmoid output layer.

3.2.3 Squeeze and Excitation Network

This architecture is based on the sample-level CNN architecture proposed in [26]. It uses one-dimensional raw audio-data as input. A basic block consists of a convolutional layer (3x3), followed by batch-normalization and max-pooling (2x2 for 2D and 3 for 1D). After that, the squeeze and excitation (SE) part is applied. The squeeze operation consists of a global average pooling to compress each channel. The excitation operation consists of two fully-connected layers (ReLU, sigmoid). Finally, the output of the basic convolutional part gets scaled with the output of the SE part by channel-wise multiplication. The squeeze and excitation network (SENet) consists of nine of these blocks, where the outputs of the last three blocks are max-pooled globally and concatenated to serve as input for two FC output layers, which use sigmoid outputs [26].

To make this architecture work for 2D input representations the frequency dimension is reduced gradually using 2D filters throughout the network.

3.2.4 Musicnn

Musicnn is a CNN designed specifically for music tagging and has been re-implemented and slightly adapted for this work. The original architecture uses Mel spectrograms with a length of three seconds and 96 Mel-bins as input [9]. The architecture is divided into three parts: *frontend*, *midend*, and *backend*. The *frontend* uses one convolutional layer with different kernel sizes motivated by music-domain knowledge. In the original work, 7x38, 7x67, 128x1, 64x1 and 32x1 filters are used. These filters are scaled relatively to the size of the frequency dimen-

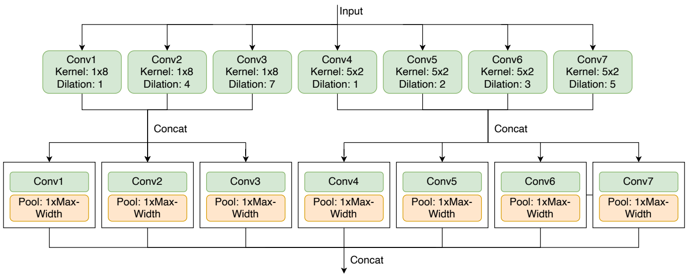
Input
Conv1 Kernel: 1x8 Dilation: 1
Conv2 Kernel: 1x8 Dilation: 4
Conv3 Kernel: 1x8 Dilation: 7
Conv4 Kernel: 5x2 Dilation: 1
Conv5 Kernel: 5x2 Dilation: 2
Conv6 Kernel: 5x2 Dilation: 3
Conv7 Kernel: 5x2 Dilation: 5
Concat
Concat
Conv1 Pool: 1xMax-Width
Conv2 Pool: 1xMax-Width
Conv3 Pool: 1xMax-Width
Conv4 Pool: 1xMax-Width
Conv5 Pool: 1xMax-Width
Conv6 Pool: 1xMax-Width
Conv7 Pool: 1xMax-Width
Concat

Figure 2. Dilated CNN frontend architecture for 2D-input

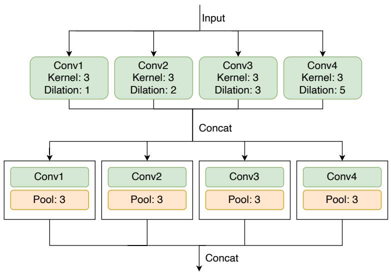
Input
Conv1 Kernel: 3 Dilation: 1
Conv2 Kernel: 3 Dilation: 2
Conv3 Kernel: 3 Dilation: 3
Conv4 Kernel: 3 Dilation: 5
Concat
Conv1 Pool: 3
Conv2 Pool: 3
Conv3 Pool: 3
Conv4 Pool: 3
Concat

Figure 3. Dilated CNN block for frontend for 1D-input

sion for the other input representations used in this work. For the one-dimensional input, the frontend architecture from [14] is used, consisting of seven 3x1 convolutions, combined with max-pooling layers. The *midend* consists of three convolutional layers (7xN kernels) supposed to extract higher-level representations from the low-level features of the frontend. In the original work, midend and frontend layers are connected by summation of each input with the previous outputs (quasi-residual connections, inspired by dense connections [27]). The output layer concatenates all previous layers of the midend. The *backend* uses temporal pooling for the final output. It combines average- and max-pooling over the time dimension of the output of the midend, concatenates their outputs and then uses two dense-layers with sigmoid outputs for the final prediction [14].

3.2.5 Dilated CNN

This section presents an extension of the musicnn using dilated convolutions in the frontend. Dilated convolutions spread out the filters over a wider area, thus achieving a larger receptive field without increasing resource consumption [21, 28, 29]. As discussed in the introduction, a large receptive field is desirable since music tags depend on global properties of the audio signal. Figure 2 displays the architectural layout of the frontend for the dilated CNN. Three convolutional layers are used to extract temporal features, while four layers span the frequency axis. Then, all outputs are concatenated and serve as input for the midend. Convolutions in the first block have 26 channels and 51 in the second one. As described for the stacked dilated convolution block in [29], the outputs are concatenated after the first block of convolutions and are used as input for the second block.

For the one-dimensional input, the musicnn frontend was adapted differently. It consists of seven sequential stacked dilated convolutions to reduce the input size sufficiently. Each of those blocks consists of four parallel convolutions stacked in two blocks, as displayed in Figure 3.

3.3 Datasets

The two music tagging datasets used in this work are: MagnaTagATune ("MTAT") [30] and MTG-Jamendo ("MTGJ") [31]. Both these datasets contain tags belonging to the three categories "genre", "instrument" and "mood" identified relevant for the tag grouping in this work [32, 33].

3.3.1 MagnaTagATune

The dataset was created using the game-with-a-purpose called TagATune [30, 34]. MTAT consists of 5 223 songs, segmented into 25 863 song-snippets with a length of 29 seconds. Each snippet comes with associated tags from a set of 188 tags. Some of the tags in the dataset are synonyms for each other, therefore similar tags such as "choir" and "choral" or "horn" and "horns" are merged. Since some tags only yield an insufficient number of training examples, only the top 50 tags by occurrence are used, which is a common practice [26, 35, 36]. This results in 17 genre-, 22 instrument-, 9 mood- and two uncategorized tags. To be consistent on the two datasets, a 60:20:20 train/validation/test data split is used. When generating the splits, snippets of the same song are always put in the same split to counteract overfitting.

3.3.2 MTG-Jamendo

MTGJ consists of royalty-free music, and contains 55 709 full-length songs from 3 565 different artists. Every song is at least 30s in length and 224s on average, resulting in 3 777 hours of audio. In order to be consistent with the MTAT, only the first 29s of each audio segment are used, which reduces training- and test-data size. In total, there are 195 different tag annotations for each song, each belonging to one of the three categories "genre", "instrument", or "mood". The set of tags has already been cleaned up and does not contain tags with the same meaning [31]. The distribution of the number of examples per tag is comparable to MTAT. Again, only the top 50 most frequent tags are used for the experiments, which is consistent with the literature [31]. There are 31 genre-, 14 instrument- and 5 mood-tags. For train-, validation, and test-splits, the first split definition from the dataset repository is used [31]. The split ratios are approximately 60:20:20. The split definition ensures that tracks of the same artists are only present in one subset and that tags, tracks, and artists are equally distributed.

4. EVALUATION

4.1 Experiment Setup

For each combination of network architecture, input representation, and dataset, a model is trained and evaluated. For MTAT seven runs with random initialization, and for MTGJ five runs are performed and results are averaged over the runs. As evaluation criteria ROC-AUC was used

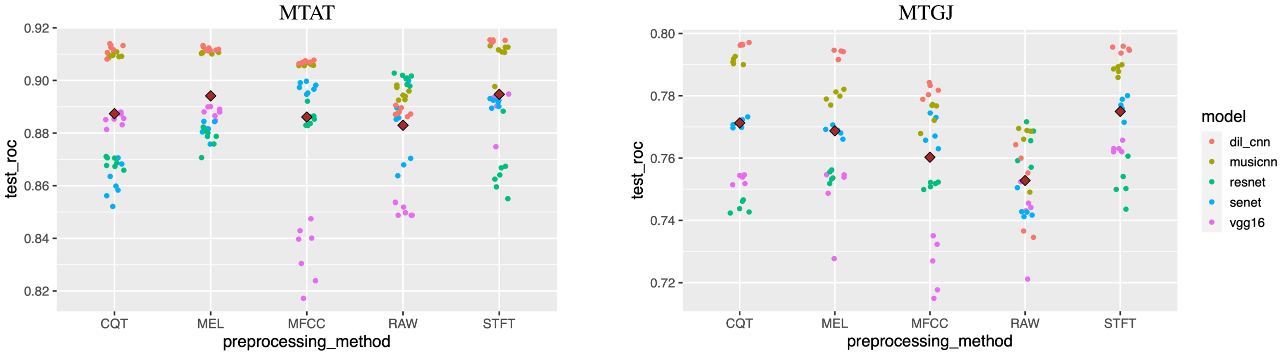

Figure 4. ROC-AUC scores for input representations and architectures on MTAT and MTGJ.

[35]. For training, early stopping was applied: training is stopped as soon as no further improvement on the validation split is achieved. As optimizer, ADAM in combination with a simple learning-rate schedule was used. Because of the smaller size of MTAT, additional refinement using stochastic gradient descent (SGD) with a reduced learning rate (1/5) was performed [37].

4.2 Results and Statistical Analysis

Figure 4 displays the ROC-AUC scores for the seven runs on MTAT (left) and the five runs on MTGJ (right) of input and architecture combinations. Different colors represent architectures, additionally, the average performance per input representation is shown in brown. The results for MTAT are on-par with state-of-the art results: dil_cnn+STFT: 0.915 v.s. 0.913 in [2]. For MTGJ, results are slightly below state-of-the-art (0.793 v.s. 0.832 in [2]), however, for compatibility reasons with MTAT in this work only 29s of each track are used v.s. full tracks in [2].

A two-way ANOVA is used to check if network architecture and/or input representation have a significant influence on the performance. Assumptions for ANOVA (independence and homoscedasticity of observations, normal distribution of residuals) are ensured: Since each run-configuration uses a different architecture and input representation combination, independence of observations is given. Homoscedasticity (equality of variances) of values is checked using the Levene test which indicates that this assumption is not met for MTAT. Therefore, a data transformation is applied to the result values on MTAT, which was not necessary for MTGJ. Residuals are found to be approximately normally distributed, which satisfies the last assumption [38].

Table 1 displays the results for the two-way ANOVA analysis. The p-values for the input representation and the architectures individually as well as combined indicate a strong influence on the performance. Due to the data transformation, the sum of squares (SS) and the mean squares (MS) for MTAT are not interpretable.

MTAT	DF	SS (E+10)	MS (E+10)	F	P
input	4	2.83	70.8	57.64	<1E-5
arch.	4	31.6	7.89	642.13	<1E-5
input&arch.	16	15.1	94.5	76.86	<1E-5
residual	150	1.84	123		
MTGJ	DF	SS (E-5)	MS (E-5)	F	P
input	4	800	200	63.61	<1E-5
arch.	4	2600	650	205.51	<1E-5
input&arch.	16	1010	60	20.04	<1E-5
Residual	100	320	3.16		

Table 1. Two-way ANOVA results on MTAT and MTGJ. SS is sum of square, MS is mean square, DF is degrees of freedom, F is f-value, and P the p-value.

	diff (t)	p-value (t)	diff	p-value
MEL-CQT	13926.88	4.92E-06	0.0068	5.11E-07
MFCC-CQT	222.24	1.0000	-0.0013	0.8135
RAW-CQT	-17197.21	1.18E-08	-0.0044	0.0024
STFT-CQT	19305.97	1.69E-10	0.0074	4.22E-08
MFCC-MEL	-13704.64	7.21E-06	-0.0080	2.05E-09
RAW-MEL	-31124.09	0.0000	-0.0112	2.85E-14
STFT-MEL	5379.09	0.2569	0.0006	0.9869
RAW-MFCC	-17419.45	7.63E-09	-0.0031	0.0651
STFT-MFCC	19083.73	2.68E-10	0.0086	1.35E-10
STFT-RAW	36503.18	0.0000	0.0118	3.12E-14

Table 2. Tukey-HSD ROC-AUC results for input representations on MTAT with transformations (t) and without.

Tukey-HSD tests are used to check for significant differences of mean values between individual input representations as well as network architectures on both datasets. Table 2 shows the results with and without the data transformation to see absolute difference between groups while being able to validate or reject those results. The overall worst performing input representation is the raw waveform. It is 0.44% worse than the CQT transformation with a p-value of 0.0024 and significantly worse for the transformed input with a p-value of 1.18e-08. Compared to the MFCCs, the raw waveform performs 0.0031% worse, but with a p-value of 0.0651, so this difference is insignificant for the untransformed input. However, with the transformed input, there is a strong significance with a p-value of 7.63e-09 that the MFCCs outperform the raw waveform to an equal extent as the CQT transformation does. Comparing the MFCCs with the CQT transformation, neither of both outperforms the other for the transformed input and the untransformed with very high p-values of 1.00 and 0.8135. As displayed before in Figure 4, the former performs better in combination with all architectures except for the VGG-16 model than the latter, where it shows a very poor performance compared to all other input representations. Mel spectrograms in average perform 0.8% better than MFCCs with a p-value of 2.05e-09 and 0.68% better than the CQT transformation with a p-value of 5.11e-07. These strong significance levels are also validated with the transformed inputs. There is no significant difference between the Mel spectrogram and the STFT with high p-values of 0.9869 and 0.2569 for untransformed and transformed inputs, respectively.

	diff (t)	p-val (t)	diff	p-value
musicnn-dil_cnn	-4834.96	0.3634	-0.0005	0.9938
resnet-dil_cnn	-84812.56	0.0000	-0.0269	0.0000
senet-dil_cnn	-78691.83	0.0000	-0.0243	0.0000
vgg16-dil_cnn	-100118.83	0.0000	-0.0370	0.0000
resnet-musicnn	-79977.60	0.0000	-0.0264	0.0000
senet-musicnn	-73856.87	0.0000	-0.0238	0.0000
vgg16-musicnn	-95283.87	0.0000	-0.0365	0.0000
senet-resnet	6120.73	0.1475	0.0026	1.93E-01
vgg16-resnet	-15306.27	4.24E-07	-0.0101	1.32E-13
vgg16-senet	-21427.00	1.95E-12	-0.0127	1.54E-14

Table 3. Tukey-HSD ROC-AUC results for network architectures on MTAT with transformations (t) and without.

Table 3 shows the results for the Tukey-HSD range test on the different architectures. VGG-16 performs significantly worse than all other models. On average, it performs 1.01% worse than ResNet and 1.27% worse than SENet with p-values of $< 1e-06$ for both transformed and untransformed scores. There is no significant difference between the latter two because the p-value is 14.75% for the transformed input. Musicnn and the dilated CNN significantly outperform all other models with $> 2.4\%$ difference. As already discussed previously, there is no significant difference between those two architectures with p-values of 99.38% and 36.34% for untransformed and transformed scores, respectively.

The results on MTGJ have been found to be consistent with those on MTAT, due to space restrictions, the tables are only available on the accompanying website. For audio representations, the raw waveform again performs worse than all other inputs, but now the significance for MFCCs outperforming the raw waveform is much stronger, with a p-value of 8.22e-05 for having a 0.75% higher ROC-score. On average, the Mel spectrogram significantly performs 0.85% better than the MFCCs, and the CQT transformation performs 1.1% better than the latter. In contrast to the previous dataset, the Mel spectrogram and the CQT transformation perform equally well. The STFT performs 0.62% better than the Mel spectrogram. However, the performance difference between STFT and CQT is not statistically significant, with a p-value of 0.1472.

Regarding architectures, there is no significant difference between the musicnn and the dilated CNN, which perform best overall. VGG-16 performs worse than other architectures, followed by ResNet and SeNet.

Additionally, the same evaluation is performed separately for the three tag categories ”genre”, ”instrument” and ”mood”. The motivation for this is the assumption that tag categories may depend on different properties of music which might be captured differently by individual input representation and network architecture combinations. On MTAT the results for genre and instrument are comparable to the overall results, while for mood the MFCC input representation performed surprisingly well, however, this could not be reproduced on MTGJ. Due to space restriction the detailed results are not provided here, but can be obtained on the accompanying website.

4.3 Discussion

The evaluation shows that both architecture and input representation significantly impact the classification results during all experiments. The STFT provides the best overall results on almost all tested configurations. This comes to the price of higher resource consumption due to the larger size of this input representation. Raw waveforms show the worst results on all configurations which might be because the architectures are originally designed for two-dimensional inputs and are adapted. Despite their small size, MFCCs perform well across almost all architectures, so they might be a good choice for tasks requiring low resource consumption. Mel spectrograms show solid performances across all architectures but are still significantly worse than the STFT. However, due to their around ten times smaller size, they are much less resource-consuming which might be the reason why they are a commonly used input representation. The dilated CNN significantly outperforms the original musicnn on two-dimensional input representations and performs worse on the raw waveform.

On MTAT the CQT transformation has the second-worst results but on MTGJ it performs equally well as the STFT. Therefore, no general conclusion can be drawn about the performance of this input representation which is comparable in terms of size to the Mel spectrogram. The evaluation on individual tag categories was inconsistent between the two datasets making further conclusions difficult. For genre tags, similar results can be observed on MTAT and MTGJ, but there is no particular trend identifiable compared to the evaluation across all tags. The results suggest that mood classification differs the most from the other tag categories. For example, MFCCs perform the second-worst for genre- and instrument classification but the second-best on mood tags. However, this was not the case for MTGJ, where MFCCs only show slightly better results than the raw waveform.

5. CONCLUSIONS

In this work, five CNN architectures and five input representations have been evaluated on two commonly used music tagging datasets. A thorough statistical analysis of the results, identifies significant performance differences between all combinations, where the combination of STFT inputs and the newly introduced dilated variant of the musicnn performs best. The evaluation on tag categories does not yield any consistent trends, but points to some interesting hypothesis, e.g., that for mood tags compressed input representations like MFCC might be sufficient.

6. ACKNOWLEDGEMENTS

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) [P33526]. For the purpose of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

7. REFERENCES

[1] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[2] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of CNN-based automatic music tagging models," in *Proceedings of the 17th Sound and Music Computing Conference (SMC)*, Toronto, Ontario, Canada, 2020.

[3] Y.-b. Yu, M.-h. Qi, Y.-f. Tang, Q.-x. Deng, F. Mai, and N. Zhaxi, "A sample-level DCNN for music autotagging," *Multimedia Tools and Applications*, vol. 80, pp. 11 459–11 469, 2021.

[4] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "A comparison of audio signal preprocessing methods for deep neural networks on music tagging," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018.

[5] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," in *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, 2018.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, 2016.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE conference on computer vision and pattern recognition (CVPR)*, Salt Lake City, Utah, USA, 2018.

[9] J. Pons and X. Serra, "Musicnn: Pre-trained convolutional neural networks for music audio tagging," in *Late Breaking and Demos of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[10] M. Huzaifah, "Comparison of time-frequency representations for environmental sound classification using convolutional neural networks," *arXiv preprint arXiv:1706.07156*, 2017.

[11] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.

[12] W.-T. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, "SpecTNT: A time-frequency transformer for music audio," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.

[13] Y. Gong, Y.-A. Chung, and J. Glass, "PSLA: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3292–3306, 2021.

[14] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[15] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[16] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, 2017.

[17] K. Muhammad, Mustaqeem, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran, G. Sannino, and V. H. C. de Albuquerque, "Human action recognition using attention based LSTM network with dilated CNN features," *Future Generation Computer Systems*, vol. 125, pp. 820–830, 2021.

[18] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[19] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, QC, Canada, 2020.

[20] B. Di Giorgi, M. Mauch, and M. Levy, "Downbeat tracking with tempo-invariant convolutional neural networks," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, QC, Canada, 2020.

[21] M. E. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proceedings of the 27th European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019.

[22] X. Zhang, Y. Zou, and W. Shi, "Dilated convolution neural network with LeakyReLU for environmental sound classification," in *Proceedings of the 22nd International Conference on Digital Signal Processing (DSP)*, London, UK, 2017.

[23] A. Ferraro, D. Bogdanov, X. Serra, J. H. Jeon, and J. Yoon, "How low can you go? Reducing frequency and time resolution in current CNN architectures for music auto-tagging," in *Proceedings of the 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, The Netherlands, 2021.

[24] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, USA, 2000.

[25] J. C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, pp. 425–434, 1991.

[26] T. Kim, J. Lee, and J. Nam, "Sample-level CNN architectures for music auto-tagging using raw waveforms," in *Proceedings of the 43th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018.

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, 2017.

[28] X. Lei, H. Pan, and X. Huang, "A dilated CNN model for image classification," *IEEE Access*, vol. 7, pp. 124 087–124 095, 2019.

[29] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, "SDC-stacked dilated convolution: A unified descriptor network for dense matching tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.

[30] E. Law, K. West, M. Mandel, M. Bay, and J. Downie, "Evaluation of algorithms using games: The case of music tagging." in *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[31] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-Jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML)*, Long Beach, CA, United States, 2019.

[32] T. Bertin-Mahieux, D. Eck, and M. Mandel, "Automatic tagging of audio: The state-of-the-art," *Machine Audition: Principles, Algorithms and Systems*, pp. 334–352, 2010. [Online]. Available: https://www.igi-global.com/gateway/book/40288

[33] G. Marques, M. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging." in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, USA, 2011.

[34] E. L. Law, L. Von Ahn, R. B. Dannenberg, and M. Crawford, "Tagatune: A game for music and sound annotation." in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007.

[35] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016.

[36] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[37] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," *arXiv preprint arXiv:1906.04972*, 2019.

[38] M. Kozak and H.-P. Piepho, "What's normal anyway? Residual plots are more telling than significance tests when checking ANOVA assumptions," *Journal of agronomy and crop science*, vol. 204, no. 1, 2018.