Using NN with Spectrogram as Input in MER: Final Report

G079 (s1804121, s1843212, s1834912)

Abstract

This project evaluates whether the usage of spectrogram with more complex CNN models, can improve results of MER. Several ways of generating spectrogram including STFT, CQT and their phase spectrograms are also adapted and tested. DEAM dataset is used in the project to provide continuous emotion annotations. Result shows CNN models can be effective to predict time-variant data , and using CQT spectrogram can be helpful for performance improvements. For accuracy, the model built with inception nodes and trained with CQT spectrograms provides very competitive results compared to results generated by LSTM.

1. Introduction

Music can affect people's emotion in various ways where simultaneous feelings come across, including hopeful or happy, angry or sad. This might be results of the interaction between sound melody or rhythm and human brain, say, a biological instinct according to (Nakada et al., 1998). In other word, listeners varies in nationalities, native languages, education backgrounds can have emotional responses toward music due to the correlation between music and human brain, even though they can hardly tell what music is. This give rise to the music emotion recognition (MER) task which has precise description given by (Barthet et al., 2013), where various contents (audio and lyrics) and context-based features (pitch, tempo and other features) have been used for features for a long time (Juslin & Laukka, 2004). Although multi-label classification problems have been taken into consideration follows (Wieczorkowska et al., 2007) to extract features (Juslin & Laukka, 2004; Wieczorkowska et al., 2007; Liu & Chen, 2015), the performance is far from satisfaction. One essential obstacle in MER is the inconsistency of emotion towards same music segments for different people, which leads to significant challenge in extracting key features affecting people's emotion. That is, it is hard to tell the reason why various emotions, which can even be contradict ones, are inspired by different people given the same segment of a specific song.

An innovative model using CNN architecture is put forward by (Liu et al., 2017) trying to solve the problem. In this approach, instead of involving complex features manually, music audio spectrograms are used as input in Convolutional Neural Network (CNN), which reserve original time and frequency information while get rid of complicated model construction. In addition, problems caused by different length of different music segments are avoided. Furthermore, since there is a process of emotional cumulation with time when people listen to music, the spectrogram can better represent the music.

Aiming to address the problems of MER task and improve its performance, we follow the step of (Liu et al., 2017), trying to make improvements and evaluate them while using spectrograms in neural network models for emotion recognition tasks. Improvements mainly focus on the generation of spectrograms and different CNN architectures. That is, 1) three types of models, which will be described in detail in section 3 might be taken into consideration to evaluate applicability of CNN for MER; 2) two ways of generating spectrograms (STFT, CQT and their phase spectrograms, which will be stated in section 2.1 will be adapted to give a better spectrogram.

In the following section, introduction of data set along with tasks will be given (section 2). Afterwards, the models used in experiments will be described in detail (section 3). To explore characteristics of models and to evaluate the effect of techniques in spectrogram generation, several experiment will be conducted (section 4). Finally, related work (section 5) which can help to give better understanding and future exploration of the work, will be also given before conclusion (section 6).

2. Data set and task

2.1. Data set

In this project the DEAM data set is used (Aljanaki et al., 2017). Different from the CAL500exp (Wang et al., 2014) data set used in (Liu et al., 2017), DEAM data set provides emotion annotations as continuous levels in Valence-Arousal (VA) space (Russell, 1980) instead of discrete emotion classes. In this context, two numbers are used respectively for valence, unpleasant to pleasant, and arousal, calm to excited, to describe one emotion. DEAM data set provides two groups of annotations for every audio clip: average value and continuous value. This project will focus on continuous values, which are given twice every second after 15 seconds from the beginning. For a 45 second audio clip, it provides 60 numbers for both arousal and valence. These values are acquired by using an interface to continuously collect feedback from the workers as one point in VA space, then computing the average between workers. The range for continuous emotion values are -1 to 1 for both valence and arousal, but are scaled linearly to [-0.5, 0.5] in order to compare with results generated by other similar projects.

The data set contains 1802 audio clips, in which 1744 clips are intended to be cut to 45 seconds by the producer, from a random starting point for each song, while the sizes of remaining ones are random. In this project, only the 1744 clips with fixed sizes are used because variable length will lead to variable sizes of spectrogram, which will bring extra complexity to the problem when using CNN for analysis. The data set do not provide separation for training set, validation set and testing set, so they have to be manually partitioned. By observing the data, it can be found there is strong correlation between valence and arousal values. Therefore, the clips are first sorted by sum of valence and arousal, then split to each set by a round-robin separator, to ensure each set have similar statistical characteristics. In this project, clips are allocated to each sets with ratio of 6:1:1 for validation, training and testing. In result, there are 1308 clips in training set, 218 clips in validation set and testing set.

Spectrograms are generated for the audio clips as input for CNN model. Two types of transforms are utilized for spectrogram generation, namely short-time Fourier transform (STFT) and constant-Q transform (CQT), of which figure 1 provides examples. Short-time Fourier transform divides the the audio clip into short pieces, then apply Fourier transform to each piece, to generate immediate frequency characteristics of the original audio clip. Similar to STFT, constant-Q transform performs partitions to the audio clip, but uses a slightly tweaked function in domain transformation. As the result, different from generated by STFT, spectrograms generated by constant-Q transform have frequency values in logarithmic scale, which can be considered to be a meaningful feature for following reasons:

1. Pitches are in logarithmic frequency scale to human. So spectrograms generated by constant-Q transform have equivalent spacing between pitches and octaves, making it easier to extract features like chromatic scales and chords.

2. It provides more frequency steps at low frequency part of the sound (lower than 3kHz), which is the main frequency range human can easily notice. Therefore, it provides more details at low frequency part and more intuitive representations of music to human's hearing.

3. It uses less frequency steps to represent a large frequency range, improving data density and reducing loss of details in compression compared to spectrogram generated by STFT with same size.

Considering computing resource available for this project, the full size of the spectrograms are chosen to be about 512x512. To achieve this size of spectrogram, FFT size is chosen to be 1024 and hop length is chosen to be 2048, resulting in spectrograms of size 485x513. For constant-Q transform, hop length is still 2048, and 48 bins are used for each octave, usually larger for practical usage, with 8 octaves in total. This provides spectrograms of size 485*384, which are still smaller than STFT spectrograms. Each spectrogram is generated in complex domain, then amplitude and phase is extracted separately, generating 2 channels. Therefore, for STFT spectrogram, the full size is 2x485x513 and for CQT spectrogram, the full size is 2x384x485.

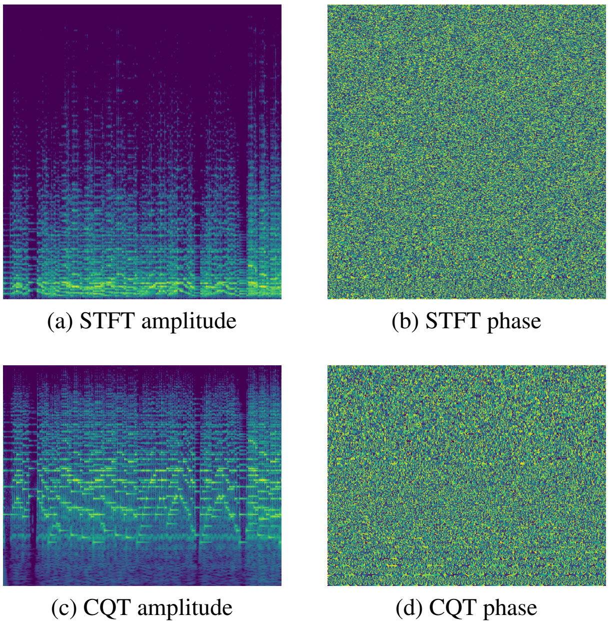
(a) STFT amplitude (b) STFT phase (c) CQT amplitude (d) CQT phase

Figure 1. List of different types of spectrograms

STFT	SQT	Code	Text
485x513	485x384	1	16x
242x256	242x192	2	4x
121x128	121x96	4	1x

Table 1. List of variant in size with STFT represents spectrogram size generated by STFT, SQT represents spectrogram size generated by SQT, code represents identifier used in code and filename, text represents identifier to be used in following sections

For spectrogams smaller than this size, there can be observable loss of details, but training complex models with full size spectrograms can be time consuming. Therefore, the full size spectrograms are resized to several sizes to provide flexibility in experiments. The varients are given in table 1.

2.2. Tasks

The first task of this project is to evaluate applicability of CNN to time-variant data. In a spectrogram, x-axis usually means time and y-axis usually means frequency. For most of popular structures of CNN, the kernels for convolution are square and all the operations have no preference on axis, which means there should be ideally no difference in perfor-

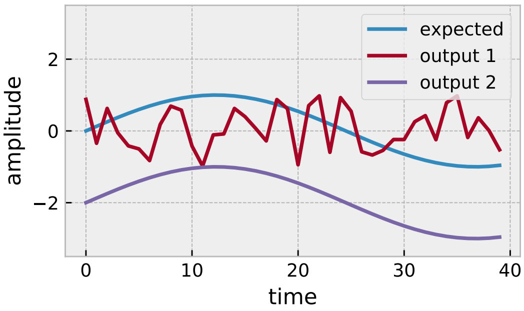

Figure 2. Examples for RMSE (lower is better) and correlation (higher is better), with output 1 showing low RMSE and low correlation, and ouput 2 showing high RMSE and high correlation

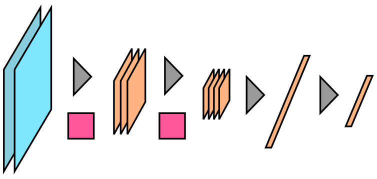

Figure 3. An example of Alex model using blue planes for input spectrograms, orange planes for generated features, and red rectangles for convolution kernels

mance when rotating some images by 90°. This is a good feature for most use cases of CNN, at the scarification of potential to have specific patterns for each axis. In this project, the performance of some CNN models will be tested, using spectorgram as input to generate time-variant outputs, specifically, emotion annotations for different timestamps.

RMSE and averaged correlation ($\bar{\rho}$) will be used to evaluate the result. In this case, RMSE can only provide the general difference between expected outputs and actual outputs, but cannot determine if the output follows the trend of input data. On the other hand, correlation provides better representation if the prediction follows the time-variant data. Figure 2 provides two examples to show the effect of the two values.

Another task is to evaluate the effect of choosing different types of spectrogram as input. As described in section 2.1, constant-Q transform seems to provide better representation of frequency characteristics. Therefore, spectrograms generated by STFT and CQT will be used as input of CNN models to check the difference in result. Also, although phase spectrograms seems to be noise for human, it will be tested if using phase spectrogram in addition to amplitude spectrogram will have effects on performance. RMSE will be the main criterion for this task and average correlation will be given for reference.

3. Methodology

Three different models are used in this project to provide comparisons between different model structures in CNN, and to provide more generalized results for CNN in spectrogram analysis. The first model uses a very basic structure which is similar to AlexNet, so it is called Alex. Figure 3 shows a simplified structure of it: use convolution with 3x3 kernel and max pooling for layers at start, then use fully connected layers to generate outputs. This model is used as baseline for performance.

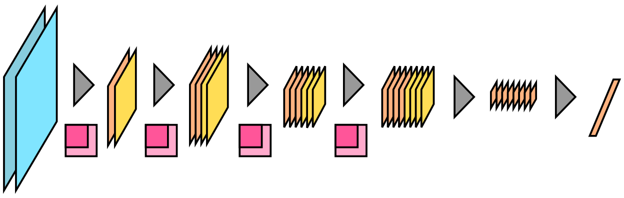

Figure 4. An example of dnception model using blue planes for input spectrograms, orange and yellow planes for features generated by different methods, and red rectangles for convolution kernels in differnt size

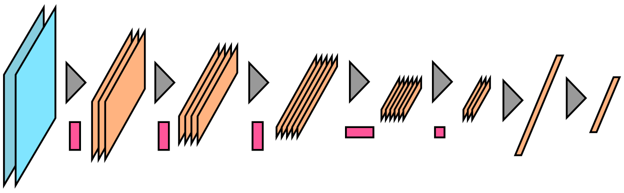

Figure 5. An example of directional model using blue planes for input spectrograms, orange planes for generated features, and red rectangles for convolution kernels

The second model uses inception nodes, which are famous for usage in GoogLeNet (Szegedy et al., 2014), to increase the depth of the network, so it is called inception model. The simplified structure is given in figure 4. In each layer, the inception node uses a combination of different methods to manipulate the data, including pooling and convolution with kernels in different sizes, then combines the outputs into different layers in output data. To generate output, every channel is converted to one number using average pooling, similar to the design in GoogLeNet, then supplied to one fully connected layer to generate outputs. In this model, to further increase the depth, pooling is applied to the data for every 2 inception layers, rather than one pooling for every convolution layer, which is used in the structure of Alex model.

The third model uses an uncommon design which is dedicated to this project, which is called directional model. It firstly uses convolution layers with 1x3 kernel and 1x3 pooling layers to reduce the height of the input map to a small number, then uses convolution layers with 3x1 kernel and 3x1 pooling layers to reduce the width to an acceptable width, after which the model have highest number of channels. Afterwards, it uses 1x1 convolution to reduce the channels, and finally uses fully connected layers to generate outputs.

The design of directional model is to explicitly keep the location of features. For input spectrograms, when reducing the height of feature map, the model is forced to extract the frequency features at one time point, without changing the length of sample. Afterwards, by reducing the width, the model is then forced to summarize changes in time. When doing width reduction, the width is kept to a descent number to store time-variant characteristics, which is the key idea of the design. At this stage, since the data has large number of channels, supplying the data to fully connected layers will generate numerous amounts of parameters, so 1x1 convolution is used for channel reduction before usage of fully connected layers for outputs. This design is similar to TDNN, but it uses 2D convolution to extract the patterns, while frequencies are treated as separate inputs in TDNN, so this design could make better use of frequency features. Also, another benefit of this design is to increase the depth of network, because 1-directional pooling uses more layers to resize the feature map to a certain size.

All the models are implemented to accept variable sizes of feature maps, to provide convenience when using spectrograms with different sizes mentioned in table 1. However, this method may lead to different behaviors for different input sizes. For example, using input map twice bigger will usually lead to extra layers in each model because it requires more pooling layers for dimension reduction and number of convolution layers is usually linked to number of pooling layers. What can be ensured in this case is the structure of each model will be the same for spectrograms with same reduction level (same row in table 1). Another variable '`size`' is introduced to control the size of model, representing the highest amount of channels. Implementation of each model will adapt to this variable dynamically, according to specific design of structures. By changing the size of models, the user can easily control the usage of VRAM and consumption of time.

4. Experiments

4.1. Baseline Model

Alex model is chosen as the baseline model for all the experiments because the structure is simple, making it fast to train and easy to understand. All of the models are built with maximum of 256 channels, then trained for 100 epochs using spectrograms generated by constant-Q transform in 2x size without phase information as input. Figure 6 shows the learning curve of executions using 3 different learning rates, with each learning rate executed twice using different random initial parameters. Numeric data of each execution are given in table 2.

Since there are no regularization methods used, training loss is reduced to less than 0.001 within 40 epochs for models with learning rate of 3e-5 and 1e-4, while it takes longer for models with smaller learning rate. For validation loss, the values converge between 0.008 to 0.010, which seems to be a small range, but actually generates significant difference in performance. According to previous results (Aljanaki et al., 2017), the test loss for most methods are about 0.09, which is relatively easy to achieve, while loss less than 0.08 can be considered very competitive. For higher learning

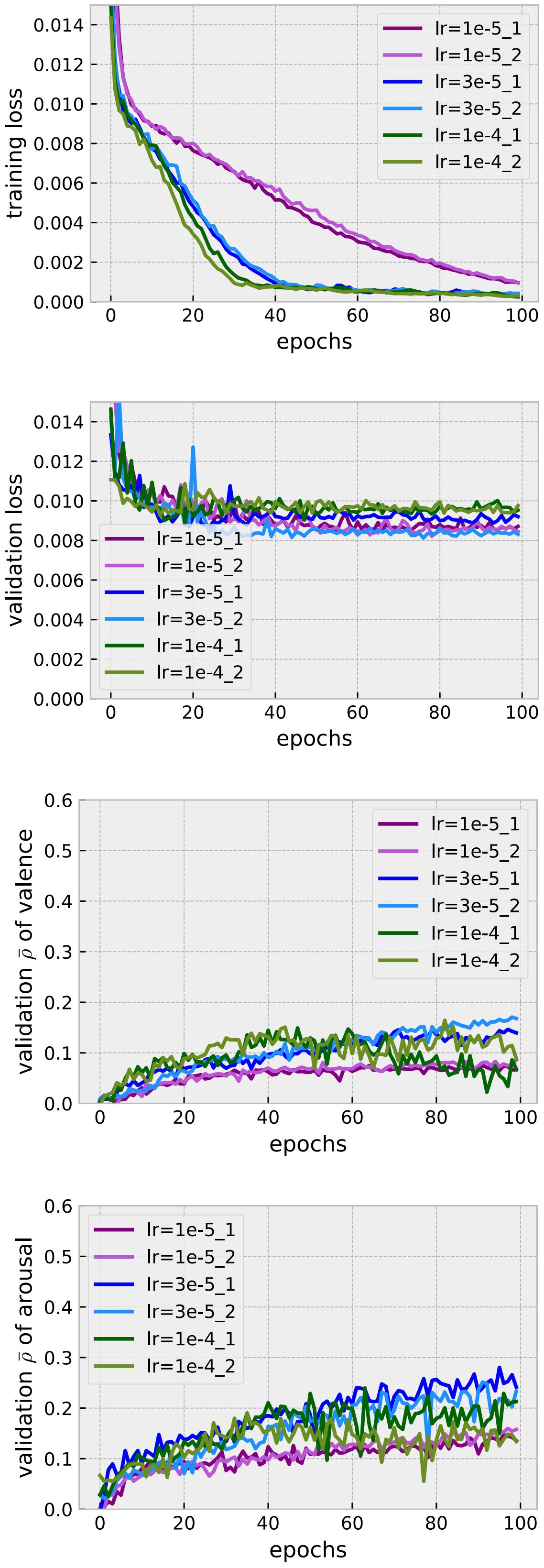

Figure 6. Learning curves of baseline model with lr representing learning rate

LR	Index	Parameters	RMSE of valence	$\bar{\rho}$ of valence	RMSE of arousal	$\bar{\rho}$ of arousal	Loss
1e-5	1	1.27M	0.088	0.094	0x081	0.129	0.0082
1e-5	2	1.27M	0.087	0.078	0.083	0.146	0.0081
3e-5	1	1.27M	0.091	0.083	0.086	0.168	0.0085
3e-5	2	1.27M	0.091	0.136	0.081	0.255	0.0085
1e-4	1	1.27M	0.090	0.080	0.089	0.133	0.0089
1e-4	2	1.27M	0.100	0.058	0.082	0.135	0.0097

Table 2. Statistics of baseline models

Model	LR	Parameters	RMSE of valence	$\bar{\rho}$ of valence	RMSE of arousal	$\bar{\rho}$ of arousal	Loss
Alex	3e-5	1.27M	0.091	0.136	0.081	0.255	0.0085
Inception	1e-4	722k	0.081	0.168	0.079	0.271	0.0069
Directional	3e-5	860k	0.092	0.123	0.079	0.299	0.0086

Table 3. Statistics of models used in section 4.2

Model	Phase	Spectrogram	LR	Parameters	RMSE of val.	$\bar{\rho}$ of val.	RMSE of aro.	$\bar{\rho}$ of aro.	Loss
Alex	Yes	STFT	3e-5	1.32M	0.105	0.046	0.096	0.223	0.0122
Alex	Yes	CQT	3e-5	1.27M	0.095	0.050	0.082	0.103	0.0096
Alex	No	STFT	3e-5	1.32M	0.094	0.067	0.090	0.238	0.0101
Alex	No	CQT	3e-5	1.27M	0.090	0.080	0.089	0.133	0.0089
Inception	Yes	STFT	1e-4	722k	0.093	0.089	0.086	0.136	0.0090
Inception	Yes	CQT	1e-4	722k	0.092	0.091	0.079	0.111	0.0085
Inception	No	STFT	1e-4	722k	0.080	0.160	0.078	0.261	0.0070
Inception	No	CQT	1e-4	722k	0.081	0.168	0.079	0.271	0.0069
Directional	Yes	STFT	3e-5	702k	0.095	0.084	0.085	0.284	0.0098
Directional	Yes	CQT	3e-5	860k	0.092	0.108	0.084	0.251	0.0089
Directional	No	STFT	3e-5	702k	0.097	0.063	0.088	0.248	0.0101
Directional	No	CQT	3e-5	860k	0.092	0.123	0.079	0.299	0.0086

Table 4. Statistics of models used in section 4.3

rates, it can be observed that the validation loss converges at larger value compared to curves with lower learning rate. One interesting fact about the curves is there is no obvious overfitting area in validation loss, even when training loss approaches a very low level compared to validation loss. This effect can be found with all the three learning rates.

For correlation results, there seems to be much oscillation in the curves and the results are more interesting. For the highest learning rate, over-fitting can be observed for both valence and arousal, reaching the maximum at about 50 epochs. For lower learning rates, the correlations are continuously growing, even when there is hardly any change in training and validation loss, which is not common. It possibly indicates the model is still learning time-variant features without changing the loss. Similar to the green curves, correlations are expected to start to overfit given more epochs, but it is not tested due to the limit of computing resources.

In following sections, Alex model with learning rate of 3e-5 will be given as baseline. Since the averaged correlation seems to have similar shapes for arousal and valence, correlation of arousal will be given in following sections as indicator of correlation performance, because it has larger magnitudes. Correlation of arousal will not be given if there is no special results.

4.2. Time-variant Analysis

In this experiment, models described in section 3 will be used to evaluate the performance when predicting time-variant data, as task 1 described in section 2.2. In this experiment, all the models use input in 2x size generated by constant-Q transform without phase spectrogram. Since directional model is designed for improved detection of variance in time, it is hypothesized that directional models will have better results in averaged correlation.

Figure 7 shows the learning curves of the models, where many differences can be observed. For loss function, the value of directional model decreases slowly compared to other models, while effect of overfitting can be found in validation loss after 70 epochs, when training loss starts to drop far from validation loss. In this case, the learning curve of directional model is special because other models do not seem to have clear effect of overfitting as is described in section 4.1, although it is usually discovered in experiments. This is possibly because directional model provides better exposure of time-variant features, making them easy to be detected like time-invariant features. On the other hand, since Alex model and inception model do not have special channels for time-variant features, such information have to be transferred into channels. Therefore, using more layers can be helpful to express the complexity of relationship. These models do not overfit possibly due to the implicitness of relationship.

Results for correlation is also interesting. The baseline model learns features in training set very fast, but the knowledge failed to be transferred into validation set. Considering the simplicity of the model, it possibly simply remembers the inputs, then provides the outputs based on its memory, rather than generating outputs based on the patterns. However, for inception model and directional model, they are considered to effectively learn the patterns because the training curve is very similar to validation curve. Another feature to be noticed is the directional model learns very fast in the beginning, but failed to keep the speed after 40 epochs. This is possibly causes by lost of details in height compression because connection in x-axis is totally ignored by using 1x3 convolution and pooling, making it difficult for further improvements when reaching certain accuracy.

One thing to be noticed is that best value of loss and correlation do not always occur at the same epoch. For loss function, it usually arrives at the bottom relatively early, then oscillate close to the value. For correlation, it sometimes keeps increasing through the entire training process. In most of experiments, it is normal to choose the parameters at the epoch with lowest validation loss. However, if the model is required to provide higher correlation, the developer need to be careful with curves of both validation loss and correlation, and in such case, use parameters in later epochs.

Generally speaking, CNN can be useful in detecting time-variant features, but it requires more complex structures like more layers or combination of different layers. The performance of directional model cannot surpass that of inception model, but it can be considered effective to extract time-variant features. In addition, the directional model provides very special learning curves, which are potential for further investigation.

4.3. Choice of spectrograms

This experiment aims to discover the effect of choice of spectrograms. As is described in section 2.1, CQT has many special features which could be beneficial when using spectrograms for music analysis. So it is hypothesized using CQT will improve performance of models. Although phase spectrogram seems noisy, it still contains important information of audio waves. So it is hypothesized using phase spectrograms can improve performance. Each model is tested with different settings of spectrograms: using spectrograms generated by STFT or CQT, and using phase spectrograms or not. When phase spectrograms are used, they will be supplied to models as an additional channel

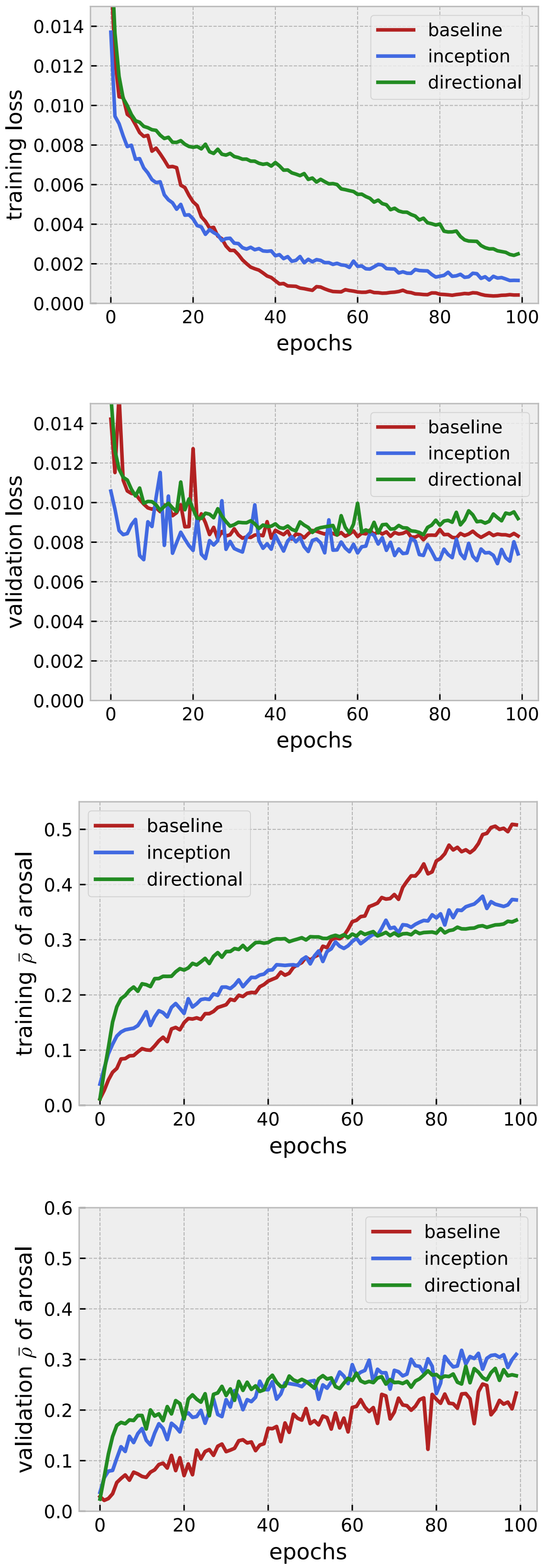

Figure 7. Learning curves of models used in section 4.2

with same size of amplitude spectrogram. The curves of validation loss are given for each model in figure 8 because it has closed relationship with test results. Other numeric data of each execution are given in table 4.

The curves seems consistent with results in previous sections: Alex model converges at relatively high loss; inception model converges at relatively low loss; directional model has similar loss as Alex model, but is the only model that has clear effect of overfitting. For different spectrograms, the result is also clear in all the models: usage of CQT spectrograms can reduce the loss, and using phase specrograms will always lead to negative effects.

Since the main difference of CQT compared to STFT is that it converts linear frequency into linear pitch, considering pitches are very important parts of music, the change in linearity might be a indicator that CNN models are more efficient when analyzing linear problems, even for models with many layers like inception mode. It can be concluded using CQT spectrograms instead of STFT spectrograms can be a good option to try in music analysis for improvements in both accuracy and training speed, since the size of CQT spectrograms are actually smaller than the ones generated using STFT.

For phase spectrograms, the result is not expected. For all the models, they all have worse performance even when accepting more data. Howerver, it still seems reasonable because the phase spectrograms are very close to noise for human. There are several reasons that phase spectrograms are not useful.

The first possibility is they are not supplied to the model correctly. When checking the phase spectrogram very carefully, some patterns can be found, the clearest of which is the phase is relatively constant for one region with high amplitude, which is consistent to the feature of waves: phase do not change for one component with fixed frequency. However, 2D convolution is not powerful to detect features like “If one pattern is discovered in one channel, the data located at same position in other channels are also important”, because each channel is given separate parameters, making it not straight forward to detect internal connections between channels. For most of problems in CNN like this, adding more layers could be an easy solution, but it seems not effective, at least within 100 epochs.

Another possibility is that phase information has very weak relationship with emotion of music. When recovering waveform from spectrograms, it is helpful with phase data. However, when phase data is not given, developers can still recover the waveform using random phases, the result of which is not so different from ones generated with phase data. Therefore, when the models are detecting patterns in phase spectrogram that cannot be transferred to validation set, it will reduce performance of the models.

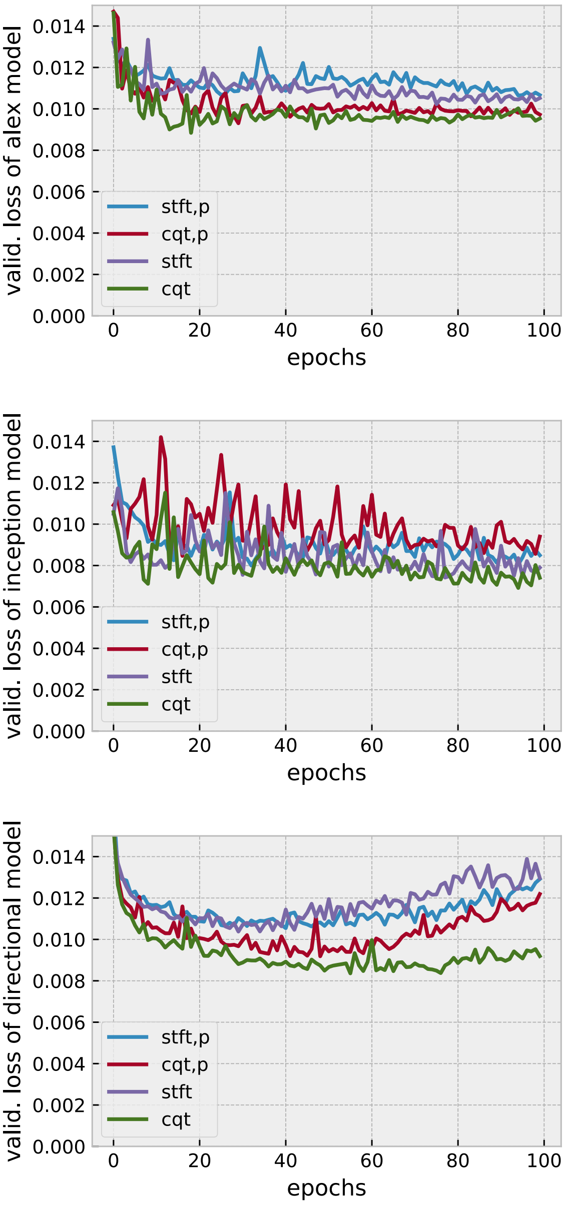

Figure 8. Validation losses of models used in section 4.3

4.4. Performance Characteristics

According to previous sections, inception model has outstanding accuracy compared to other models. However, the accuracy does not come for free. Figure 9 shows the time consumption of each model. It can be observed the inception model takes at least 4 times the time of other models. If the use case does not have high requirements for accuracy, it is reasonable to use Alex model for descent loss, or to use direction model for descent correlation.

5. Related work

Liu et al. mentioned using complex models will not always improve the performance (Liu et al., 2017). In this project, it seems inception model, which has relatively high

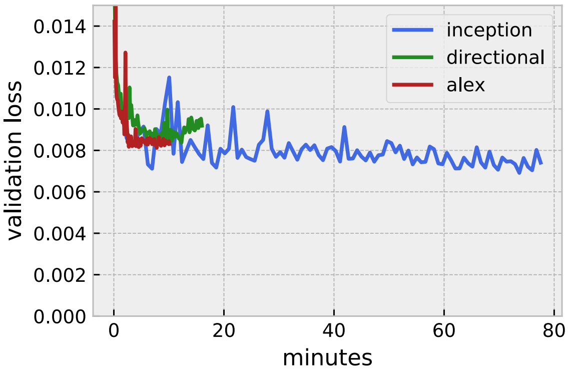

Figure 9. Validation loss in figure 7 using time as x-axis

depth, clearly provides better results compared to Alex model, which is similar to the model used in the paper mentioned. By adding more layers to Alex model, many of added parameters could be useless, because the size of dataset is relatively small. However, using inception nodes and global average pooling is famous for increasing the depth of network, without introducing much parameters, which could make it more efficient in performance.

According to the description of DEAM (Aljanaki et al., 2017), the best results achieved 0.8 in RMSE in both valence and arousal, 0.20 and 0.35 in averaged correlation for valence and arousal respectively, which are usually acquired using RLSTM (Bi-directional Long-Short Term Memory) (Weninger et al., 2013). It's not too surprising because LSTM models are known for ability to detect relationship in time. What is surprising is the results generated in this project is actually very competitive. According to table 3, the inception model achieved 0.8 in RMSE in both valence and arousal, 0.27 and 0.17 in averaged correlation for valence and arousal respectively, which is slightly worse than best results, but still better than most of teams. Since there is much space for hyperparameter tweaks like learning rate and weight decay, it is promising to use CNN model to achieve similar accuracy as LSTM models. Considering there is no previous research results generated using CNN based on DEAM dataset, it could also be a reference of design choices and performance results of CNN models.

For future research, a suggestion is to train the model for more epochs. As is described in section 4.1, using high learning rate seems to reduce accuracy of the model, while the model can not be trained to optimal correlation when learning rate is reduced. Every model in this project is trained for 100 epochs due to the limit of computing resource. Therefore, a relatively large learning rate is used to let the model finish training of correlation at the sacrifice of accuracy. With more time and more resource allowed, the models can be trained for more epochs with lower learning rate, which could be promising to generate better results.

Since LSTM models are dominant in analysis of sequential data for its performance, it is still the first choice for better results. This project can also provide suggestions to improve LSTM models, like usage of CQT spectrograms. In addition, the developers can try to use models provided by this project as encoders before supplying the spectrograms to LSTM models. This structure can have two benefits: One is to help extract graphical features that are hard to be detedcted by LSTM models. Another is to reduce the length of sequence, hence reducing the difficulty in development of LSTM models.

6. Conclusions

In conclusion, the applicability of using CNN in MER is proved, while it is beneficial to use CNN with complex structures like inception model. In addition, it is suggested to use CQT spectrograms instead of STFT spectrograms due to its improvements in both accuracy and training speed. However, due to the potential reasons including limited power in detecting inter-channel connections and weak relations between phase information and emotion of music, the performances of phase spectrograms fail to reach the expectation. For future research, it is suggested to test the model with more epochs, and to try models using a combination of CNN and LSTM for better accuracy.

References

Aljanaki, Anna, Yang, Yi-Hsuan, and Soleymani, Mohammad. Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3):1–22, 03 2017. doi: 10.1371/journal.pone.0173392. URL https://doi.org/10.1371/journal.pone.0173392.

Barthet, Mathieu, Fazekas, György, and Sandler, Mark. Music emotion recognition: From content- to context-based models. In Aramaki, Mitsuko, Barthet, Mathieu, Kronland-Martinet, Richard, and Ystad, Sølvi (eds.), *From Sounds to Music and Emotions*, pp. 228–252, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41248-6. URL https://link.springer.com/chapter/10.1007/978-3-642-41248-6_13.

Juslin, Patrik and Laukka, Petri. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33:217–238, 09 2004. doi: 10.1080/0929821042000317813. URL https://www.tandfonline.com/doi/abs/10.1080/0929821042000317813.

Liu, Shuhua (Monica) and Chen, Jiun-Hung. An empirical study of empty prediction of multi-label classification. *Expert Systems with Applications*, 42(13):5567 – 5579, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.01.024. URL http://www.sciencedirect.com/science/article/pii/S0957417415000391.

Liu, Xin, Chen, Qingcai, Wu, Xiangping, Liu, Yan, and Liu, Yang. CNN based music emotion classification. *CoRR*, abs/1704.05665, 2017. URL http://arxiv.org/abs/1704.05665.

Nakada, T, Fujii, Y, Suzuki, K, and Kwee, IL. "Musical brain" revealed by high-field (3 Tesla) functional MRI. *Neuroreport*, 9(17):3853–3856, December 1998. ISSN 0959-4965. doi: 10.1097/00001756-199812010-00016. URL https://doi.org/10.1097/00001756-199812010-00016.

Russell, James. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 12 1980. doi: 10.1037/h0077714.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott E., Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1409.html#SzegedyLJSRAEVR14.

Wang, S., Wang, J., Yang, Y., and Wang, H. Towards time-varying music auto-tagging based on cal500 expansion. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2014. doi: 10.1109/ICME.2014.6890290. URL https://ieeexplore.ieee.org/abstract/document/6890290.

Weninger, Felix, Eyben, Florian, and Schuller, Björn. The tum approach to the mediaeval music emotion task using generic affective audio features. In *Proceedings MediaEval 2013 Workshop, Barcelona, Spain*, 2013. URL https://mediatum.ub.tum.de/doc/1189713/file.pdf.

Wieczorkowska, Alicja, Synak, Piotr, and Ras, Zbigniew. *Multi-Label Classification of Emotions in Music*, volume 35, pp. 307–315. 07 2007. doi: 10.1007/3-540-33521-8_30. URL https://link.springer.com/chapter/10.1007/3-540-33521-8_30.