Predicting Perceived Dissonance of Piano Chords Using a Chord-Class Invariant CNN and Deep Layered Learning

Juliette Dubois
ENSTA ParisTech
jdubois@ensta.fr

Anders Elowsson
KTH Royal Institute of Technology
anderselowsson@gmail.com

Anders Friberg
KTH Royal Institute of Technology
afriberg@kth.se

ABSTRACT

This paper presents a convolutional neural network (CNN) able to predict the perceived dissonance of piano chords. Ratings of dissonance for short audio excerpts were combined from two different datasets and groups of listeners. The CNN uses two branches in a directed acyclic graph (DAG). The first branch receives input from a pitch estimation algorithm, restructured into a pitch chroma. The second branch analyses interactions between close partials, known to affect our perception of dissonance and roughness. The analysis is pitch invariant in both branches, facilitated by convolution across log-frequency and octave-wide max-pooling. Ensemble learning was used to improve the accuracy of the predictions. The coefficient of determination (R^2) between rating and predictions are close to 0.7 in a cross-validation test of the combined dataset. The system significantly outperforms recent computational models. An ablation study tested the impact of the pitch chroma and partial analysis branches separately, concluding that the deep layered learning approach with a pitch chroma was driving the high performance.

1. INTRODUCTION

The concept of dissonance has a long history. However, the experimental study of dissonance dates back only to the middle of the 20th century. Both its definition and causes are still subject to discussion. In early studies [1], consonance is defined as a synonym for beautiful or euphonious. Sethares [2]) proposed a more general definition of dissonant intervals: they sound rough, unpleasant, tense and unresolved.

According to Terhardt [3], musical consonance consists both of sensory consonance and harmony. Similar to this explanation, Parncutt [4] considers the existence of two forms of consonance, one being a result of psychoacoustical factors (sensory consonance) and the other one relating to the musical experience and the cultural environment (musical or cultural consonance). Dissonance can thus be divided into sensory dissonance and musical dissonance. Musical dissonance depends on our expectation and musical culture, which makes it difficult to evaluate [5]. In [6], several acoustic factors are isolated and the influence of each one is evaluated separately.

Dissonance is closely related to another sensory concept, *roughness*. Roughness applies both for music and natural sounds, and occurs when two frequencies played together produce a beating. According to Vassilakis and Kendall (2010) [7], sensory dissonance is highly correlated to roughness.

Various sensory dissonance models were proposed since the middle of the 20th century. Based on Helmoltz' theory, Plomp and Levelt [1] conducted an experiment using pure sine waves intervals. The result of this study is a set of dissonance curves for a range of frequencies and for different frequency differences. Sethares [8] gives a parametrization of these curves, resulting in a computational model of dissonance in several cases: for two or several sine waves of different amplitudes, for one complex tone, for two notes from the same timbre.

Vassilakis developed a more precise model which estimates the contribution depending on the partial amplitudes and the amplitude fluctuation [9]. This computational model, including a specific signal processing method for extracting the partials, is available online [10].

These models are closely connected, and revolve around the same core concept of bandwidth and proximity of partial frequencies. Other approaches have also been suggested: see Zwicker and Fastl [11], or Kameoka and Kuriyagawa [12]

Schön [13] conducted an experiment in which listeners rated the dissonance of a range of chords played on a piano. Dyads (chords with two notes) and triads (chords with three notes) were played and listeners were asked to rate the dissonance for each chord. The experiment showed that dissonance is easy to rate with a rather high agreement and thus could be considered as a relevant feature for describing music from a perceptual point of view [14]. The rating data from [13] together with the data from [15] were used in the current model.

Perceptual features of music, such as perceived dissonance, *speed*, *pulse clarity*, and *performed dynamics* have received an increasing interest in recent years. They have been studied both as a group ([14,16,17]) and through dedicated models ([18–20]). The trend has been towards data driven architectures, foregoing the feature extraction step. Another trend is that of ”deep layered learning” models in MIR, as defined by Elowsson [21]. Such models use an intermediate target to extract a representation accounting for the inherent organization of music. The strategy has been applied for, e.g., beat tracking [22] and instrument recognition [23] in the past. In this study, we show how pitch estimates from a previous machine learning model can be reshaped and fed as input for predicting dissonance.

1.1 Overview of the article

In Section 2, we describe the two datasets and the process for merging them. Section 3 is focused on the input representation, detailing how it was extracted for each of the two branches of the CNN. In Section 4, the design of the CNN is described, as well as the methodology used for ensemble learning and the parameter search. Section 5 presents the evaluations methodology and the results. It also includes a small ablation study and a comparison with a previous model of dissonance. Section 6 offers conclusions and a discussion of the results.

2. DATASETS

A total of 390 chords were gathered, coming from two different experiments [13, 15]. In these two experiments, the listeners were asked to rate the dissonance of recorded piano chords. A definition of dissonance was given to the listeners. In [15], the consonance was defined as "the musical pleasantness or attractiveness of a sound". In particular, "if a sound is relatively unpleasant or unattractive, it is referred to as dissonant". In [13], the following definition was given: "dissonant intervals are often described as rough, unpleasant, tense and unresolved". Both definitions referred to the unpleasantness of a sound. However, the description from [13] was more precise and already includes the fact that only intervals were studied. The definitions were close enough to give reason to believe that the same feature was evaluated.

2.1 First dataset

The first dataset (D_1) comes from the experiment conducted in [13]. It contains 92 samples of 0.5 second each, created with a sampled piano in Ableton Live. Two kinds of chords were played, consisting of either two notes (dyads) or three notes (triads). The dyads were either centered around middle C or one fifth above, ranging from unison to a major tenth. The same process was used for the triads. In total, the dataset contained 34 dyads and 58 triads.

Thirty-two listeners were asked to evaluate the sound from "not dissonant at all" to "completely dissonant", using a web interface. The listeners' musical background varied but were mostly on an amateur level with an average practice time of 4 hours per week. The inter-rater reliability as estimated by Cronbach's alpha was 0.95.

2.2 Second dataset

The second dataset (D_2) contains 298 sound examples [15]. Each sound example was recorded from a piano and has a length of approximately 2 seconds. In this dataset, there are 12 dyads, 66 triads and 220 tetrads (chords with four notes). The frequencies were adjusted so that the mean of the fundamental frequencies was middle C (263 Hz). The pitches follow a just intonation ratio, differing from the standard equal-tempered tuning used in D_1. Thirty musically trained and untrained listeners from Vienna and Singapore rated all the examples. The inter-rater reliability as estimated by the average intraclass correlation coefficient (ICC) ranged from 0.96 to 0.99. The ratings were averaged across all listeners.

2.3 Merged dataset

We used the average listener rating of dissonance of each chord as a target for our experiment. In D_1, the dissonance ranged from 0 to 40, 40 being the most dissonant. In D_2, the dissonance ranged from 1 to 4, 1 being the most dissonant. Therefore, the ratings of the latter dataset were first inverted by multiplication with -1. The listener ratings *(A)* were then normalized according to

$$A_{normalized} = \frac{A - min(A)}{max(A) - min(A)}. \tag{1}$$

.

After normalization, the most consonant chord had a rating of 0 and the most dissonant chord had a rating of 1. The input data were also normalized. (See Section 3).

3. NETWORK INPUT

Two input representations were extracted from the audio files: the constant-Q transform (CQT) spectrum and a pitch chroma. These representations aim to catch different aspects of the audio file. The representation from the CQT can capture spectral aspects of the audio, such as the distance between partials, whereas the pitch chroma represents the audio at a higher level corresponding (ideally) to the actual notes that were played.

3.1 Pitch chroma

A pitch chroma was extracted from a Pitchogram representation, as illustrated in Fig. 1. To extract the pitch chroma, the implementation from [24] was applied to the audio files for first extracting a Pitchogram representation. This representation has a resolution of 1 cent/bin across pitch and a frame length of 5.8 ms. The Pitchogram was thresholded at 2 to remove lower noisy traces of pitch. Then, a Hanning window of width 141 was used to filter across pitch. In our initial model, we then extracted activations across pitch at semitone-spaced intervals (12 bins/octave). This gave unsatisfying results, presumably due to the out-of-sync spacing with the just intonation ratios chords in D_2. Therefore, pitch activations were instead extracted in intervals of 25 cents (48 bins/octave), ranging between MIDI pitch 25 and 103. The mean activation across time for each pitch bin was then computed, using activations from time frames 20-70. The output of this filtering will be referred to as the *pitch vector*.

A pitch chroma vector, ranging an octave, was computed from the pitch vector by taking the average activation wrapped across octaves. Three chroma vectors were stacked across pitch as shown in Fig. 1. The top 6 semitones and bottom 6 semitones were then removed. This stacked pitch

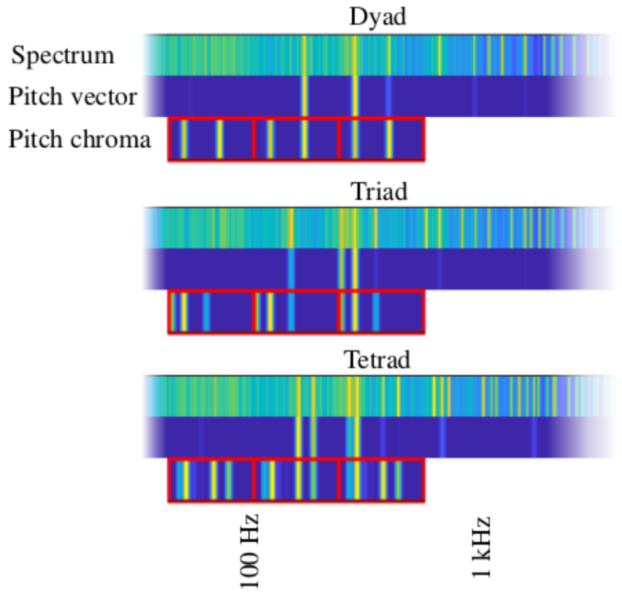


Figure 1: The log-frequency spectrum, computed pitch vector, and pitch chroma for three chords from dataset D_2. The chords are a dyad, a triad, and a tetrad. As shown, the pitch tracking preprocessing accurately identify the fundamental frequencies. The three stacked pitch chroma octaves are indicated with red rectangles.

chroma vector of size $[1 \times 96]$ will be referred to as the *pitch chroma* in this paper.

3.2 Normalization of the inputs

For each music example i the pitch chroma and the CQT were then normalized, using the same formula as in Section 2.3:

$$A_{i,normalized} = \frac{A_i - min(A_i)}{max(A_i) - min(A_i)}. \quad (2)$$

3.3 CQT vector

To extract a spectral input representation, the recordings were processed with the built-in MATLAB function `cqt`, which uses nonstationary Gabor frames (see [25], [26]). This produced a spectrogram representation with logarithmically spaced frequency bins. We used 60 bins per octave, and the range of the CQT is six octaves, 80 Hz - 5.1 kHz. The mean magnitude across time was then computed for each frequency bin, using only the first half of each audio file (the second half of the audio file has a lesser contribution from higher harmonics). The last preprocessing stage was to compute the log magnitude of this mean:

$$CQT_{mean} = 20\, log_{10}(CQT). \quad (3)$$

The resulting vector contains 360 values.

4. NETWORK DESIGN

4.1 Architecture

The architecture of the network is shown in Fig. 2.

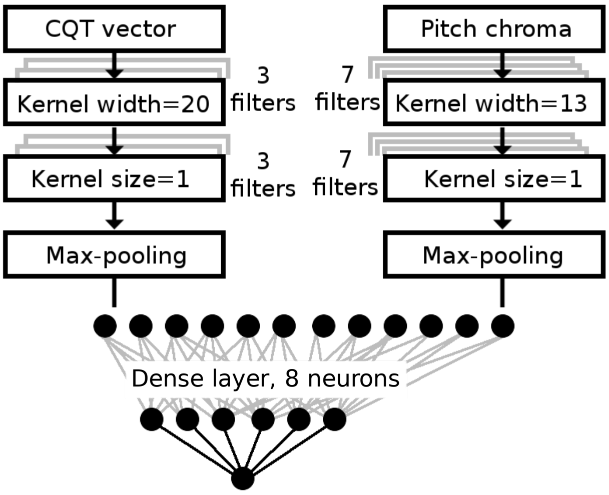


Figure 2: Architecture of the neural network

As shown, the CNN is a directed acyclic graph divided into two branches, where one branch processes the input from the CQT and the other branch processes the pitch chroma representation.

The first layer in the CQT-branch was a convolutional layer of size $[20 \times 3]$. The kernel was designed to have a width that covers at least two close partial peaks in the spectrum. The aim of the layer is to capture the interactions between peaks that are adjacent across frequency, as this should convey roughness as outlined in Section 1.

The subsequent convolutional layer also had $[3]$ filters, but each filter had a width of one, therefore only extending across depth. A max-pooling filter was then applied across frequency to capture the most relevant partial interaction in different bands. The max-pooling filter with a width 60 and no stride was then applied.

The branch processing the pitch chroma also had two convolutional layers. The first layer operating across pitch had a size of $[49 \times 7]$. Since the edges were not padded during processing, the processing shrinks the pitch chroma to a width of 48, corresponding to pitches within the same octave. This was followed by a layer of width 1 that extended across depth. A max-pooling filter of width 48 (the full range) was then applied to each filter output. Up until this processing stage, the chroma branch of the CNN has operated in a pitch class equivariant way – the same operation has been applied to all pitch classes, with the pitch class displacement intact across chroma (hence *equivariance* and not *invariance*). Through the pooling operation across chroma, the *pitch class equivariance* is transformed into *pitch class invariance*. However, since the activations after pooling will relate only to the intervals of concurrent tones, the system is best defined as having a *chord class invariant* architecture. After max-pooling, the branches are concatenated (see Fig 2) and passed to a dense layer with 8 neurons. The output layer consisted of a single neuron.

The activation functions after the CQT and the dense layer were rectified linear units (ReLUs), and the activation functions in the whole chroma branch were leaky ReLUs:

$$f(x) = \begin{cases} x & x \geq 0 \\ 0.2\,x & x < 0 \end{cases} \tag{4}$$

The network was trained for 40 epochs, using the RMS propagation optimizer and the mean square error as a loss function.

4.2 Ensemble learning

We used ensemble learning, training multiple instances of each network and averaging their predictions. A total of five models were employed in the ensemble, all using the same architecture but with varying random initialization of their parameters. The random initialization of neural networks will decorrelate the errors of the various models [27]. The average prediction from the different models can then be expected to provide better estimates than when randomly choosing one of them [28]. A similar strategy has been used before for training a model to predict perceived performed dynamics in music [20].

4.3 Parameter search

A wide range of possible settings was tried with a parameter sweep. For each setting, a model was trained and evaluated with 5-fold cross-validation.

The explored parameters, with tested parameter variations in parenthesis, were: the size of the kernel for the filter operating on the CQT (20 - 10 - 30), the number of filters for the first convolutional layer operating on the CQT and pitch chroma (6 - 7 - 8 for the CQT and 2 - 3 - 4 for the pitch chroma), the pooling size for the CQT branch, and the number of neurons in the dense layer (7 - 8 - 9). The pooling size of 60 was chosen, which corresponds to the range of an octave.

We also tried to include an additional feature inserted at the dense layer, which indicated the dataset of each chord example (1 or 2). This feature did not improve the performance of the network, and thus it was not kept.

5. RESULTS

5.1 Train and Test conditions

As there were only few data available, the system could easily overfit. To compensate for this lack of data, cross-validation was implemented, and the two datasets were also combined. These two techniques aim at adding variety in the learning set.

Thus, different methods were used to evaluate the performance of the network:

- A – Cross-validation on both datasets combined.
- B_1 – Cross-validation within dataset D_1.
- B_2 – Cross-validation within dataset D_2.
- C – Train on dataset D_2 and test on dataset D_1.

We used 10-fold cross-validation for A, B_1 and B_2, splitting the datasets into ten folds (nine folds for training and one fold for validation).

For the evaluation C, the system was trained with the dataset D_2 and evaluated on the dataset D_1. Given that the two datasets have rather different characteristics (timbre, tuning, and number of notes in the chords) this evaluation condition is more challenging. Since the dataset D_1 consists of so few examples, the opposite evaluation condition (train on D_1, evaluate on D_2) was not explored.

The metric used to compare the prediction and the rated dissonance was the coefficient of determination, R^2, computed as the squared Pearson correlation coefficient, including an intercept.

Confidence intervals (95 %) were computed, based on the variation in results between different test runs. The 5 test runs were then sampled with replacement 10000 times and the distribution of mean correlations calculated.

5.2 Main Results

The coefficient of determination R^2 across the different test conditions (A, B_1, B_2, and C) are shown in Table 1.

Test condition	Average R^2	95 % CI
A	0.631	0.622 - 0.634
B_1	0.612	0.590 - 0.634
B_2	0.644	0.621 - 0.665
C	0.583	0.561 - 0.600

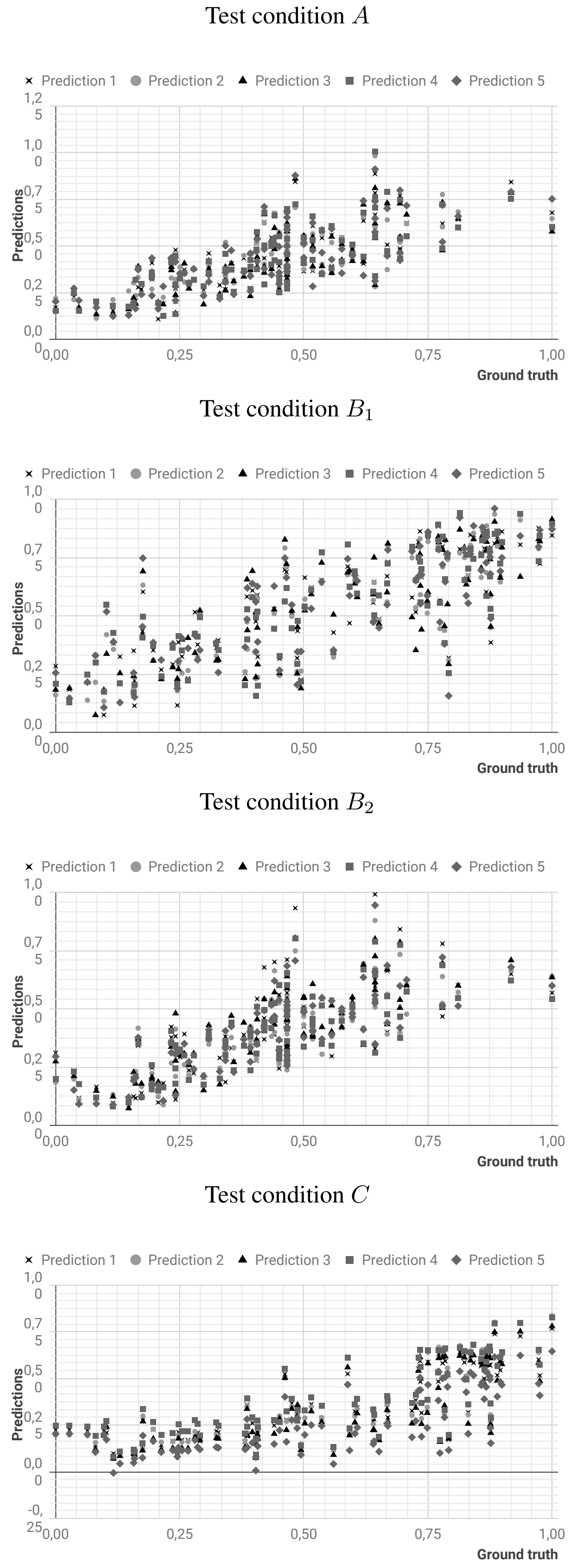
Table 1: Coefficient of determination $R2$ for the different test conditions, including 95% confidence intervals computed across the different test runs.

The predicted dissonance with respect to the target value for each music example is plotted in Fig. 3. Each point in this figure corresponds to the value of dissonance for one music example. The x-coordinate of the point is the target value of dissonance and the y-coordinate is the prediction of dissonance. For each test condition, the predictions of five different test runs (respectively called prediction 1 - 5 in the figure) are shown.

The cross-fold validation for each dataset gives comparably good results. The test on D_2 gives better results, which could be explained by the number of sample in each dataset: D_2 has four times more sample than D_1.

The cross-fold validation when combining both datasets yields better results than cross-fold validation for every single dataset. When combining both datasets, the network has a few more examples to train on. This also adds a lot a variety in the training sets, given all the differences listed before. The size of D_2 added with D_1 is not very different than the size of D_2, but the performance in A is significantly better than in B_2. This may indicate that by adding variety in the training set, the network learns much better.

The test C_1 is the only test in which the network learns on only one dataset, and this configuration gives the worse performance. Presumably, the network overfits on D_2 and cannot generalize well enough in order to predict better the values from D_1.



Test condition A

Test condition B_1

Test condition B_2

Test condition C

Figure 3: Predictions in relation to ground truth annotations for the different test conditions

Test Condition	Average R^2	95 % CI
Network using only the pitch chroma		
A	0.618	0.610 - 0.624
B_1	0.641	0.629 - 0.649
B_2	0.628	0.620 - 0.633
C	0.607	0.589 - 0.625
Network using only the CQT vector		
A	0.417	0.409 - 0.426
B_1	0.304	0.281 - 0.327
B_2	0.460	0.439 - 0.484
C	0.180	0.139 - 0.221

Table 2: Coefficient of determination R^2 for the network with only one input: CQT or pitch chroma, including 95% confidence intervals computed across the different test runs.

5.3 Contribution from each input source and branch

In order to evaluate the importance of the pitch chroma and CQT vector for performance, we also ran the full experiment using only the pitch chroma in a single branch or only the CQT vector in a single branch. Other than this, the same settings were used during training, and the same metric used for testing. The results are shown in Table 2.

Using pitch chroma as the only input consistently gave better results than when only using the spectral input from the CQT. Furthermore, the pitch chroma had better results than the combined main model (reaching outside of the 95 % CIs) for test conditions B_1 and C. The CQT vector had particularly low results for test condition C. This condition tests the ability of the architecture to generalize since the system is trained on one dataset and tested on another with, presumably, different characteristics pertaining to, e.g., timbre. The results confirms that the deep-layered learning approach to MIR [21], in this paper using transfer learning of equivariant feature maps, can yield significantly better results than end-to-end learning for small datasets.

5.4 Comparison with a Computational Model

In this section, comparison of the performances of the system with a state of the art model is presented. The most recent model was proposed by Vassilakis [9] and does not use machine learning. It was already implemented as the function `mirroughness` in the MIR toolbox [29], which is the implementation we used for the comparison.

With this function, a dissonance value is given for each unit of time, which was not directly comparable with the single value given by the listeners. Considering that a human listener would not take the length of the recording into account, we chose to take the mean of the five highest values.

With this method, a dissonance value was computed for each music example. The squared correlation coefficient was then computed between this dissonance and the target dissonance. The results are shown in Table 3.

The R^2 score obtained here is lower than the performances obtained in the article presenting the model [9]. This could be explained by the fact that the computational model was tested and adapted to synthetic sine waves, whereas the audio files in this experiment came from a sampled piano. The timbre of the piano presumably increases the complexity of the sound and reduce the accuracy of the model.

Dataset	R^2
D_1	0.17
D_2	0.34

Table 3: Coefficient of determination $R2$ for the computational model for each dataset

6. CONCLUSION AND DISCUSSION

A model using a convolutional neural network was developed for predicting the dissonance in recordings of piano chords. The model achieved better results than previous computational models, even though there were few samples in the datasets.

The two datasets differ in at least four ways:

- The chords were played with two different piano models, producing differences in, e.g., timbre.
- One dataset was performed with equal-tempered chords, and one with just intonation ratio. The model, therefore, had to handle micro-tuning deviations and how they affect dissonance.
- One main difference is the polyphony level of the chords: D_1 has no tetrads whereas these constitute more than two-thirds of D_2.
- The two datasets were rated by two different groups of listeners. Therefore, it can be expected that random variations between preferences in the two groups gave annotations that varied in complex ways.

We conclude that the tests validate the potential of intermediate targets accounting for the inherent organization of music. The ”deep layered learning” approach [21] using only the pitch chroma branch gave significantly better results than when using only the spectral CQT vector branch. In particular, a comparison between the results for test condition B_2 and C underlines the pitch chroma-only model's high generalization capability. The R^2 only fell slightly (0.628 – 0.607 = 0.021) when testing on an unseen dataset instead of using cross-validation. For the main model with both branches, the results fell more between these test conditions (0.644 – 0.583 = 0.061).

During development, we tested a few different architectures, with fewer learnable parameters in total, but those architectures gave lower results. It seems like the architecture allowed for a fairly high amount of learnable parameters in relation to the low number of ground truth data points.

In future work, a wider range of architectures could be tried, reflecting insights gained from the small ablation study. The pitch chroma branch can be designed as the only branch, exploring improvements related to, e.g., pitch resolution, depth, and pooling. A related study [30] has showed that it possible to compute several chroma within the network instead of as a preprocessing step, each chroma focusing on different octaves. That study also indicated that average pooling across octaves for key-class invariance can give better results than max-pooling. It could be useful to analyze the weights in the kernel on the trained network. This could make it easier to understand the characteristics selected by the network. The network could also be trained with a much bigger dataset, using several repetitions of the same chord class or using chords from higher and lower octaves.

Acknowledgments

The authors appreciate the help of Daniel L. Bowling who made the data of his work available.

7. REFERENCES

[1] R. Plomp and W. J. M. Levelt, “Tonal consonance and critical bandwidth,” *The Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965. [Online]. Available: https://doi.org/10.1121/1.1909741

[2] W. A. Sethares, “Relating tuning and timbre,” 1992. [Online]. Available: https://web.archive.org/web/20100610092432/http://eceserv0.ece.wisc.edu/~sethares/consemi.html

[3] E. Terhardt, “The concept of musical consonance: A link between music and psychoacoustics,” *Music Perception: An Interdisciplinary Journal*, vol. 1, no. 3, pp. 276–295, 1984. [Online]. Available: http://mp.ucpress.edu/content/1/3/276

[4] R. Parncutt, *Harmony : A Psychoacoustical Approach*. Springer, 1989.

[5] A. Porres, “Dissonance model toolbox in pure data,” in *Pure Data Convention Weimar - Berlin*, 2011.

[6] J. Mcdermott, A. J Lehr, and A. J Oxenham, “Individual differences reveal the basis of consonance,” vol. 20, pp. 1035–41, 06 2010.

[7] *Psychoacoustic and cognitive aspects of auditory roughness: definitions, models, and applications*, vol. 7527, 2010. [Online]. Available: https://doi.org/10.1117/12.845457

[8] W. Sethares, “Local consonance and the relationship between timbre and scale,” *The Journal of the Acoustical Society of America*, vol. 94, 09 1993.

[9] P. N. Vassilakis, “Perceptual and physical properties of amplitude fluctuation and their musical significance,” Ph.D. dissertation, University of California, Los Angeles, 2001.

[10] "Spectral and rougness analysis of sound signals," http://musicalgorithms.ewu.edu/algorithms/roughness.html. [Online]. Available: http://musicalgorithms.ewu.edu/algorithms/roughness.html

[11] E. Z. H. Fastl, *Psychoacoustics*. Springer, 2009.

[12] A. Kameoka and M. Kuriyagawa, "Consonance theory part ii: Consonance of complex tones and its calculation method," *The Journal of the Acoustical Society of America*, vol. 45, no. 6, pp. 1460–1469, 1969. [Online]. Available: https://doi.org/10.1121/1.1911624

[13] R. Schön, "Vertical dissonance as an alternative harmonic-related perceptual feature," KTH DM 2906 Individual Course in Media Technology, Tech. Rep., 2017.

[14] A. Friberg, E. Schoonderwaldt, A. Hedblad, M. Fabiani, and A. Elowsson, "Using listener-based perceptual features as intermediate representations in music information retrieval," *The Journal of the Acoustical Society of America*, vol. 136, no. 4, pp. 1951–1963, 2014.

[15] D. L. Bowling, D. Purves, and K. Z. Gill, "Vocal similarity predicts the relative attraction of musical chords," *Proceedings of the National Academy of Sciences*, vol. 115, no. 1, pp. 216–221, 2018. [Online]. Available: http://www.pnas.org/content/115/1/216

[16] A. Friberg and A. Hedblad, "A comparison of perceptual ratings and computed audio features," in *Proceedings of the SMC 2011 - 8th Sound and Music Computing Conference*, 2011, pp. 122–127. [Online]. Available: http://www.smc-conference.org/smc11/papers/smc2011_163.pdf

[17] A. Aljanaki and M. Soleymani, "A data-driven approach to mid-level perceptual musical feature modeling," *CoRR*, vol. abs/1806.04903, 2018. [Online]. Available: http://arxiv.org/abs/1806.04903

[18] F. A. M. G. . P. J. Elowsson, A., "Modelling the speed of music using features from harmonic/percussive separated audio," in *14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013, pp. 481–486.

[19] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari, "Multi-feature modeling of pulse clarity: Design, validation, and optimization," in *In Proceedings of the International Symposium on Music Information Retrieval*, 2008.

[20] A. Elowsson and A. Friberg, "Predicting the perception of performed dynamics in music audio with ensemble learning," *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 2224–2242, 2017. [Online]. Available: https://doi.org/10.1121/1.4978245

[21] A. Elowsson, "Deep layered learning in MIR," *CoRR*, vol. abs/1804.07297, 2018. [Online]. Available: http://arxiv.org/abs/1804.07297

[22] M. D. F. Krebs, S. Böck and G. Widmer, "Downbeat tracking using beat synchronous features with recurrent neural networks," in *ISMIR*, 2016, pp. 129–135.

[23] Y.-N. Hung and Y.-H. Yang, "Frame-level instrument recognition by timbre and pitch," in *ISMIR*, 2018, pp. 135–142.

[24] A. Elowsson, "Polyphonic pitch tracking with deep layered learning," *CoRR*, vol. abs/1804.02918, 2018. [Online]. Available: http://arxiv.org/abs/1804.02918

[25] *Matlab documentation on cqt*, https://se.mathworks.com/help/wavelet/gs/non-stationary-gabor-frames.html. [Online]. Available: https://se.mathworks.com/help/wavelet/gs/non-stationary-gabor-frames.html

[26] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Drfler, "A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Jan 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17112

[27] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, Oct 1990.

[28] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, Third 2006.

[29] O. Lartillot, P. Toiviainen, and T. Eerola, "A matlab toolbox for music information retrieval," in *Data Analysis, Machine Learning and Applications*, C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 261–268.

[30] A. Elowsson and A. Friberg, "Modeling music modality with a key-class invariant pitch chroma cnn," manuscript submitted for publication, 2019.