POLITECNICO DI TORINO

MASTER OF DATA SCIENCE AND ENGINEERING

MASTER'S THESIS

EXPLAINABILITY METHODS IN MUSIC EMOTION RECOGNITION

Supervisor
Prof. Cristina Rottondi

Candidate
Giacomo Zuliani

DECEMBER 2025

Acknowledgements

I would like to use these few lines to express my gratitude to some special people who made this journey possible.

First and foremost, I want to thank my parents, Stefano and Fabiana, for laying the foundations for my growth and for their constant support throughout my life. My siblings, Chiara and Marco, have also been a source of encouragement, motivation and fun along the way.

I am also grateful to my old friends and new ones alike — having people you can rely on, and who inspire you to grow, makes all the difference. In particular, I want to thank my longtime friends, the Gremlins, for their enduring friendship and the laughter we share, my flatmates, whose company made everyday life more enjoyable, my coursemates, who have been both a reliable support in my studies and wonderful companions along the way, and my volleyball teammates, for the fun, friendship, and balance they brought to this period of my life.

A special thanks goes to Agnese, a wonderful person who helped me enormously and stood by my side in the final part of this journey.

Finally, I would like to thank Professor Cristina Rottondi for guiding me through this thesis, for her steady support, and for her kindness and availability throughout our work together.

Giacomo Zuliani
Turin, December 2025

Abstract

This thesis explores how explainability can be introduced into Music Emotion Recognition (MER) models, which are usually hard to interpret despite their good performance. While many deep learning models can predict the emotional content of music with high accuracy, they often work as black boxes, giving little to no information about how they reach their conclusions. The goal of this work is to make these models more understandable, especially for users who might want to exploit them not just as tools, but also to learn something from them.

To do this, the thesis develops and tests two different approaches. The first one is based on musical features—some taken from the literature, and others introduced as a novel contribution. It starts from an existing deep learning framework that uses mid-level features like melodic or rhythmic descriptors to explain predictions, and then expands it by adding simpler, more intuitive features like chords or notes that could be easier to interpret and possibly helpful to composers or researchers.

The second approach instead focuses on raw audio data. Here, the idea is to make the model's internal reasoning perceptible through sound. Using a Vision Transformer (ViT) trained on spectrograms and Layer-wise Relevance Propagation (LRP), this method creates a modified version of the original music where the most relevant parts for the prediction of the task Happy vs. Sad are made louder, allowing the listener to hear which segments influenced the classification most.

Even if the two approaches are different, they aim at the same objective: making MER models more transparent and easier to interact with. The hope is that this kind of explainability can help both in research and in creative applications, giving users a better grasp of the models' internal reasoning processes.

Code and audio demos:
https://github.com/giacomozu/Explainability-Methods-in-Music-Emotion-Recognition

Contents

1 Introduction

1.1 Motivation and Problem Statement

Music has always played a central role in human emotional expression. Whether through melody, rhythm, or harmony, musical structures seem to resonate with deeply rooted affective patterns. In recent years, this expressive capacity has become the focus of computational models aimed at understanding or even predicting emotional responses to music. This research field, known as MER, has seen rapid development, especially with the rise of deep learning methods.

Yet, as predictive accuracy has improved, a significant problem remains: the opacity of these models. Most modern MER systems behave as black boxes, providing outputs (e.g., "happy" or "sad") without offering any insight into the reasoning that led to those decisions. This lack of interpretability is not a trivial issue. It affects the reliability of these systems and, more importantly, prevents users from using them as tools for musical understanding, composition, or research.

This thesis tackles this issue by posing the following research question: How can we build models for MER that not only perform well but also reveal something about how music and emotion are related? What kinds of explanation might be useful, for example, to a composer trying to shape the affective impact of a piece, or to a researcher studying musical communication?

1.2 Objectives of the Study

The main goal of this work is to explore different strategies for making MER models more interpretable. Two complementary directions are developed throughout the thesis.

The first is grounded in musical structure. It focuses on using features that carry an intuitive musical meaning, like melodic shape, chord sequences, or rhythmic stability, as a bridge between audio and emotion. The idea is that if the model reasons through familiar musical concepts, then its predictions can be more easily

interpreted by humans.

The second direction takes a more perceptual route. Instead of looking at symbolic representations, it works directly with audio (more precisely, with spectrograms) and uses attribution methods to identify which parts of the input were most important for the model's decision. These relevance maps are then turned into sound using a sonification process, so that the explanation becomes something that can be heard rather than just visualized.

Although these two approaches differ in methodology, they share the same objective: making the internal logic of MER models more accessible. By combining structural and perceptual perspectives, the thesis aims to explore how explainability can not only improve trust in the model, but also enrich our understanding of music itself.

1.3 Anticipated Findings.

While a detailed discussion is left to the later chapters, it is useful to highlight here some of the main lessons that emerged from the experiments. In the feature-based route, combining harmonic and timbral cues turned out to be more effective than using either source alone. In particular, concatenating chromagrams with spectrograms provided a clear boost over single representations, confirming that emotional perception in music depends on both pitch structure and fine spectral detail. Linear regression models with mid-level and symbolic features also proved to be surprisingly strong: their stability and interpretability made them preferable to shallow neural networks in most cases, even when the latter offered more flexibility.

On the perceptual side, the ViT classifier reached a solid level of accuracy on the Happy vs. Sad task, and the use of LRP revealed that rhythmic regularity was one of the strongest cues exploited by the model. Through sonification, these patterns became clearly audible, showing how the system tended to associate steady pulses with happiness and darker, sustained textures with sadness. Taken together, these results suggest that explainability is not only possible but can actively shape modelling choices: richer feature sets and perceptual attributions both pointed to musically meaningful cues, offering insights that go beyond raw metrics.

1.4 Structure of the Thesis

The thesis develops along two parallel experimental directions, both aimed at improving the interpretability of MER systems. The first direction focuses on structured musical features such as chroma, mid-level perceptual descriptors, and chord encodings as inputs for interpretable models. The second direction explores perceptual approaches based on deep learning, with a Vision Transformer trained on spectrograms and analyzed through attribution and sonification techniques.

These two paths are introduced and elaborated progressively across the chapters, allowing for a coherent and comparative understanding of their goals, methodologies, and outcomes.

After introducing the theoretical foundations in **Chapter 2**, where the key concepts of MER, explainable AI (XAI), and audio-derived representations are discussed, the thesis continues in **Chapter 3** with a review of the most relevant related work, including both traditional and deep learning approaches to MER, the use of mid-level features, and recent developments in explainability for audio tasks.

Chapter 4 presents the methodological framework that supports both experimental directions of the thesis. It details the datasets used, the feature extraction processes, the architecture of the models, and the explainability techniques adopted, including SHAP and Layer-wise Relevance Propagation (LRP). This chapter also outlines the sonification pipeline developed to transform model attributions into perceptually meaningful audio.

Chapter 5 is dedicated to the experiments and the discussion. It is divided into two main sections: the first one focuses on models built using symbolic and mid-level musical features, analyzing their predictive performance and interpretability; the second one covers perceptual models based on Vision Transformers trained on spectrograms, and explores their behavior using LRP and audio-based sonification. Quantitative and qualitative results are presented for both paths and their findings are discussed, highlighting the benefits and limitations of each.

Chapter 6 summarizes the main contributions of the thesis and outlines possible directions for future research in both explainability and music emotion modeling.

2 Theoretical Background

2.1 Fundamentals of MER

MER is concerned with the task of predicting the emotional content of a musical excerpt based on its audio signal. Emotions in this context can be expressed using either categorical labels (e.g., happy, sad, angry) or continuous dimensions (such as valence and arousal), depending on the type of annotation provided in the dataset.

One of the main challenges in MER stems from the inherently subjective nature of emotional perception in music, which is influenced by individual experience, cultural background, and listening context. Despite this, researchers have identified shared patterns that link certain musical elements to consistent emotional responses, allowing for the development of computational models capable of making meaningful predictions. These models are typically trained on datasets where human annotators have provided ratings for short musical excerpts based on how the music made them feel.

MER lies at the intersection of computer science, psychology, and musicology, and its applications are numerous: from personalized music recommendation systems to affect-aware interactive systems and even therapeutic interventions where music is used as an emotional regulator.

2.2 Theoretical Music Features

The theoretical underpinnings of music provide a structured framework for understanding the elements that constitute musical compositions. These include notes, intervals, chords, and chord progressions—each of which has well-defined mathematical properties that can be encoded computationally.

2.2.1 Notes

Notes are the atomic units of musical language, each corresponding to a particular pitch determined by frequency. In Western music, twelve distinct pitches form

the chromatic scale, named with the letters A–G together with their sharp or flat variants. The frequency of a note in equal-temperament tuning is

$$f_n = f_0 \cdot 2^{n/12},$$

where f_0 is a reference frequency (typically 440 Hz for A4) and n is the number of semitone steps from that reference.

2.2.2 Intervals

An interval denotes the distance between two notes, measured in semitones. Intervals play a key role in both melodic and harmonic contexts. For instance, the interval between C and G (a perfect fifth) spans seven semitones and corresponds to a frequency ratio of 3:2. Understanding and encoding intervals allows for a compact representation of melodic contours and harmonic structures.

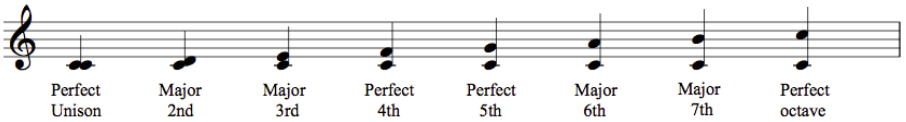


Figure 1: Visual representation of melodic intervals starting from C (Do). Each note pair illustrates a common interval type, ranging from the perfect unison to the perfect octave.

2.2.3 Octave

Two notes separated by twelve semitone steps—i.e. when n differs by 12—are said to be *one octave* apart. Because $2^{12/12} = 2$, an octave corresponds to a doubling (or halving) of frequency: for example, A3 is 220 Hz, A4 is 440 Hz, and A5 is 880 Hz. Although their frequencies differ, octave-related notes share the same letter name and are perceived as musically equivalent, a property that underlies octave-folding in chromagrams.

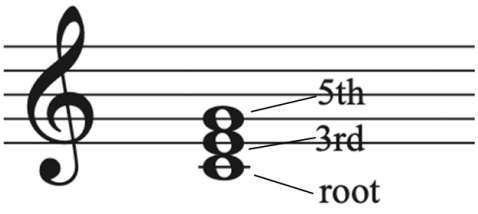
2.2.4 Chords

Chords are combinations of notes played simultaneously, typically built by stacking thirds on top of a root note. For example, a C major triad consists of the notes C, E, and G. Chords can be described using interval-based encoding, which lends itself well to symbolic representations in computational models.

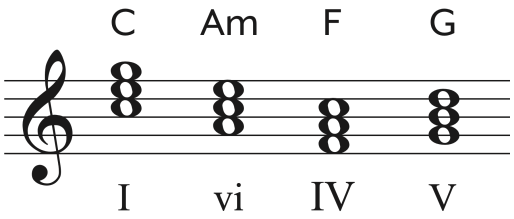
2.2.5 Chord Progressions

Chord progressions are sequences of chords that establish harmonic context and direction. They are often described using Roman numerals based on scale degrees (e.g., I–IV–V–I in C major corresponds to C, F, G, and C). These progressions can be transposed and generalized across keys, making them ideal candidates for analysis in music information retrieval and emotion modeling tasks.

By encoding notes, chords, and progressions in structured formats, it becomes possible to analyze their contribution to emotional content computationally, forming the basis for the feature-based branch of this thesis.



(a) The chord of C major, composed of root note (C), major third (E), and perfect fifth (G). The symbol on the left is a treble clef, which indicates the pitch range of the notation.



(b) A common chord progression (C, Am, F, G). Roman numerals indicate scale degrees relative to the key.

Figure 2: Visual representation of a chord and a chord progression. Each chord in these examples is formed by 3 notes with an interval of 2 semitones between them.

2.3 Audio-Derived Representations

To enable machine learning algorithms to process audio, we must first convert the waveform into representations that retain essential information while being struc-

tured in a format compatible with computational models.

2.3.1 Spectrogram

A *spectrogram* visualises how the spectral content of a signal evolves over time. The discrete-time waveform $x[n]$ is first segmented into overlapping frames of N_{FFT} samples, each new frame starting H samples after the previous one (the *hop size*). Before transformation, every frame is multiplied point-wise by an analysis window $w[n]$[1]. The Short-Time Fourier Transform (STFT) of the signal is then computed as:

$$X[m,k] = \sum_{n=0}^{N_{\text{FFT}}-1} x[n+mH]\, w[n]\, \mathrm{e}^{-i2\pi kn/N_{\text{FFT}}}, \qquad \begin{cases} m \in \{0,\ldots,M-1\}, & \text{(frame index)} \\ k \in \{0,\ldots,\frac{N_{\text{FFT}}}{2}\}, & \text{(frequency bin)} \end{cases} \tag{1}$$

The complex coefficients are converted to a *log-magnitude spectrogram* by taking their magnitude and expressing it in decibels relative to the maximum magnitude in the whole STFT matrix:

$$S[m,k] = 20\log_{10}\left(\frac{|X[m,k]|}{\|X\|_{\max}}\right). \tag{2}$$

This operation can be, e.g., performed by leveraging the function amplitude_to_db() from librosa, called with `ref=np.max`. Setting the reference amplitude to the maximum value in the spectrogram (ref=np.max) ensures that 0 dB corresponds to the most energetic point, and all other values are negative. More importantly, it makes the dynamic range relative to the signal itself.

This is particularly useful in MER, where absolute loudness can vary a lot across tracks. What matters more is the relative distribution of energy over time and frequency. By normalizing each spectrogram to its own maximum, we focus on these internal patterns—highlighting timbral and rhythmic cues—rather than being

[1] A tapering function—typically Hann or Hamming—that smoothly brings the frame amplitude to zero at its ends, thereby reducing spectral leakage.

misled by global amplitude differences. The resulting 2-D array S is displayed as an image whose horizontal axis represents time (frames m), vertical axis represents frequency (bins k), and pixel intensity encodes relative energy in decibels—brighter pixels correspond to higher energy. Because it preserves both timbral and rhythmic cues, the spectrogram is a fundamental input representation for downstream tasks such as MER.

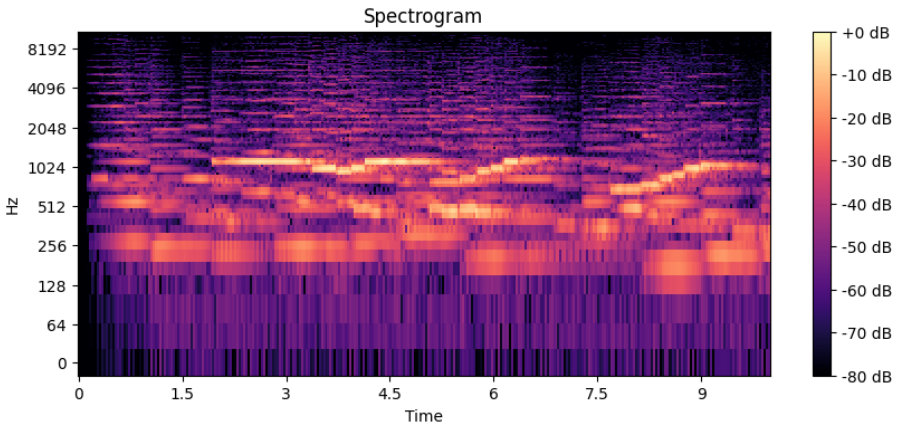


Figure 3: Example of a **log-magnitude** spectrogram. The x-axis represents time in seconds, while the y-axis shows frequency in Hz on a logarithmic scale. Colour intensity encodes signal amplitude in decibels (dB), with brighter regions indicating higher energy. Lower frequencies are concentrated near the bottom, while harmonics and higher-frequency components appear toward the top.

2.3.2 Chromagram

A *chromagram* (or *chroma feature*) condenses the spectrum into the twelve pitch classes in the equal-tempered scale (C, C♯, D, . . . , B), with notes from every octave merged into those same twelve bins. Starting from the same STFT magnitude $|X[m,k]|$ defined in Eq. (1), we aggregate the energy of every frequency bin whose pitch maps to the same pitch class.

Let $K = \frac{N_{\text{FFT}}}{2}$ be the number of frequency bins considered in the STFT. Then define a binary mapping matrix $H \in \mathbb{R}^{12\times(K+1)}$, where:

$$H_{p,k} = \begin{cases} 1, & \text{if the centre frequency of bin } k \text{ belongs to pitch class } p, \\ 0, & \text{otherwise,} \end{cases} \qquad p \in \{0,\dots,11\}.$$

be a $12 \times (K+1)$ binary mapping matrix. The (unnormalised) chroma matrix is

$$C[p, m] \;=\; \sum_{k=0}^{K} H_{p,k}\, |X[m, k]|. \tag{3}$$

By discarding octave[2] information and retaining only pitch classes, chromagrams emphasise harmonic progressions and melodic contours, which are highly relevant for tasks such as key detection, chord recognition, and, in our case, music-emotion prediction.

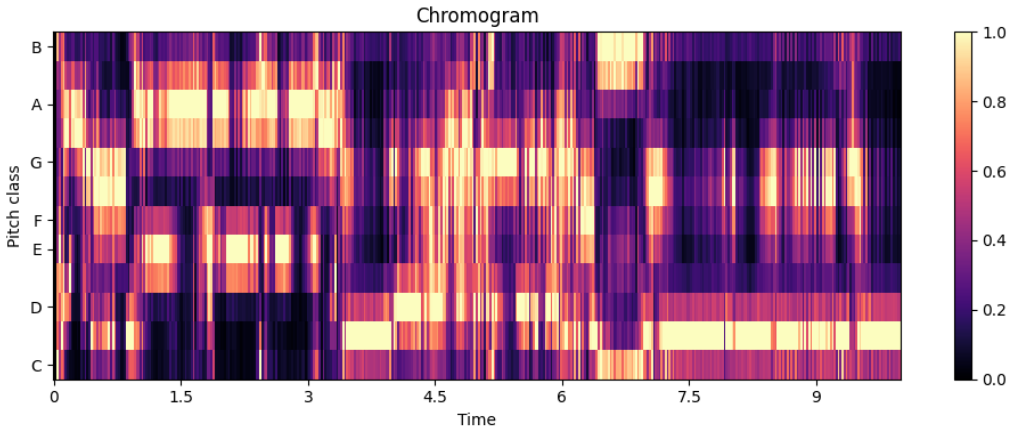


Figure 4: Example of a **normalised chromagram**. The x-axis represents time in seconds; the y-axis lists the twelve pitch classes, octave-folded. Colour intensity encodes the relative strength of each pitch class (0–1 after normalisation), with brighter cells indicating higher energy. Chromagrams capture harmonic and tonal structure while abstracting away absolute pitch, making them a powerful representation for music-emotion analysis.

In Section 5 we will see that also the 12-bin semplification of the chromogram is used for training. This simply consists in the 12 average values of the Pitches from C to B across the whole time axis.

2.4 Overview of Model Architectures

This section introduces the three main types of models used throughout this thesis: Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and linear models. Each of these was chosen depending on the type of input data being considered and the goals of each experiment, especially considering the trade-off between

[2]See Section 2.2.3.

accuracy and interpretability.

2.4.1 Convolutional Neural Networks (CNNs)

CNNs are amongst the most common choice when working with data that can be represented as images, which is the case when we turn audio signals into spectrograms or chromagrams. These models are made up of convolutional layers that scan the input image with filters and learn to detect patterns like edges, textures, or more complex shapes as the network goes deeper **[Choi et al., 2017; LeCun et al., 1998]**. In the experiments reported in this thesis CNNs were used both for regression tasks (predicting continuous emotion scores) and for classification (assigning an emotion label to a clip).

Each convolutional block is usually followed by ReLU activations, batch normalization, pooling layers, and sometimes dropout to prevent overfitting.

- **ReLU (Rectified Linear Unit):** an activation function defined as $f(x) = \max(0, x)$. It introduces non-linearity into the network, allowing the model to learn complex patterns while being computationally efficient.
- **Batch Normalization:** a normalization technique applied to intermediate layers of the network. It standardizes the inputs of each layer to have zero mean and unit variance, stabilizing and accelerating training while also acting as a regularizer.
- **Pooling Layers:** operations (such as max pooling or average pooling) that reduce the spatial dimensionality of feature maps. This helps decrease the number of parameters, improve computational efficiency, and provide a form of translation invariance.
- **Dropout:** a regularization method where, during training, a fraction of neurons is randomly "dropped" (i.e., temporarily set to zero). This prevents co-adaptation of neurons and reduces the risk of overfitting, leading to better generalization.

Even though these models typically exhibit good performance, they are not particularly easy to interpret, since their internal workings are quite opaque unless we use some post-hoc explainability tools like SHAP (see Section 2.5.1) or LRP (see Section 2.5.2).

The way features are learned in CNNs follows a hierarchy. In the early layers filters might respond to a single onset or to a couple of harmonics stacked vertically, while later layers can recognize timbral textures or recurring rhythmic patterns. This multi-level process mirrors how humans perceive music, combining small details into bigger structures. Pooling plays a role here too: by reducing the size of the feature maps it forces the network to keep only the most relevant information and makes the detection more robust to small shifts in time or frequency. In practice, this means a note or an onset can still be picked up even if it happens a little earlier or later than expected.

Normalization and dropout are more on the technical side, but they matter in practice because they stabilize training and help the model not to just memorize the training data. This is especially important in MER, where datasets are limited in size and can have a lot of variability from one song to another.

At the end of the convolutional stack, the feature maps are flattened and fed into dense layers. These fully connected layers essentially combine all the features into the final prediction: either a set of probabilities through a softmax when doing classification (e.g. happy vs. sad), or continuous values when doing regression. This pipeline, which is sketched in Figure 5, is the standard way CNNs are used in MIR tasks.

Of course CNNs also have their weak points. Since the filters are local, the network does not directly capture long-term structure, which is very relevant in music. Also, even if we can sometimes look at a filter and see that it matches something like an onset, the overall reasoning process of the network is still a black box. This is why CNNs in this thesis are complemented with explainability tools. Still, their efficiency and their ability to capture meaningful short- and mid-term patterns make them a good baseline, and also a useful comparison point for newer models such as Vision Transformers.

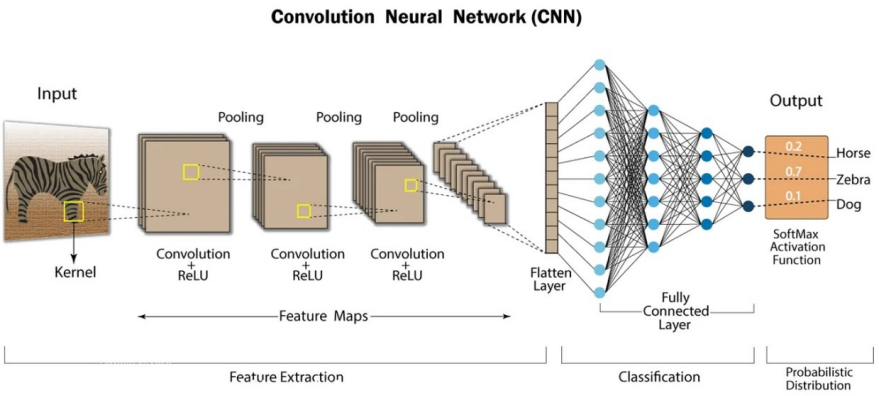


Figure 5: Example of a CNN pipeline for classification. Convolutions and pooling progressively reduce the input into smaller feature maps, which are then passed to dense layers that output the final prediction. The figure is meant to show the overall flow rather than the exact details of any particular model. *Reprinted from Haque, 2023.*

2.4.2 Vision Transformers (ViTs)

Transformers have been a huge breakthrough in NLP, and recently they have been adapted for vision tasks too. ViTs, introduced by **[Dosovitskiy et al., 2021]**, treat an image as a sequence of patches and process them using self-attention mechanisms. Interestingly, ViTs can model long-range dependencies and complex interactions between different parts of the image: in this case, different time-frequency regions of a spectrogram.

In this thesis, a pre-trained ViT model on spectrogram images was leveraged and fine-tuned, with the goal of classifying emotional content (Happy vs Sad). The ViT does not have built-in biases like local connectivity or translation invariance, so it tends to be more flexible but also needs more data to learn well. To make training feasible, pre-trained weights from ImageNet **[Deng et al., 2009]** were used and spectrograms were refined to fit the input requirements.

One of the reasons suggesting the usage of a ViT in this context was not just to test performance, but also to apply LRP and visualize what parts of the spectrogram were most important for the model's decisions. This enabled to associate good performance with some form of interpretability, even though these models are complex and require more computational resources.

Compared to CNNs, ViTs do not rely on convolutions and pooling but instead break

the spectrogram into a set of patches that are treated like tokens. Each patch is linearly projected into an embedding and positional encodings are added so that the model still knows the order of patches. A special `[class]` token is added at the beginning of the sequence. This token does not correspond to any actual patch of the spectrogram: instead, it is learned during training to gather information from all the other tokens through self-attention, so that in the end it represents a global summary of the input. This `[class]` token is then passed through a small multilayer perceptron (MLP) head that outputs the class prediction. The key difference from CNNs is that the self-attention mechanism can directly connect distant parts of the input, so the model can, for example, link a bass onset with high-frequency harmonics further away in time.

This flexibility comes at a cost: training a ViT from scratch usually requires very large datasets. That is why pre-training on ImageNet and then fine-tuning on spectrograms was essential here. Another point is that ViTs, being heavy models, need more computational resources than CNNs. Still, they are particularly interesting in this thesis because the self-attention and the use of attribution methods such as LRP make it possible to open the "black box" a little more and connect the decision process with meaningful musical cues.

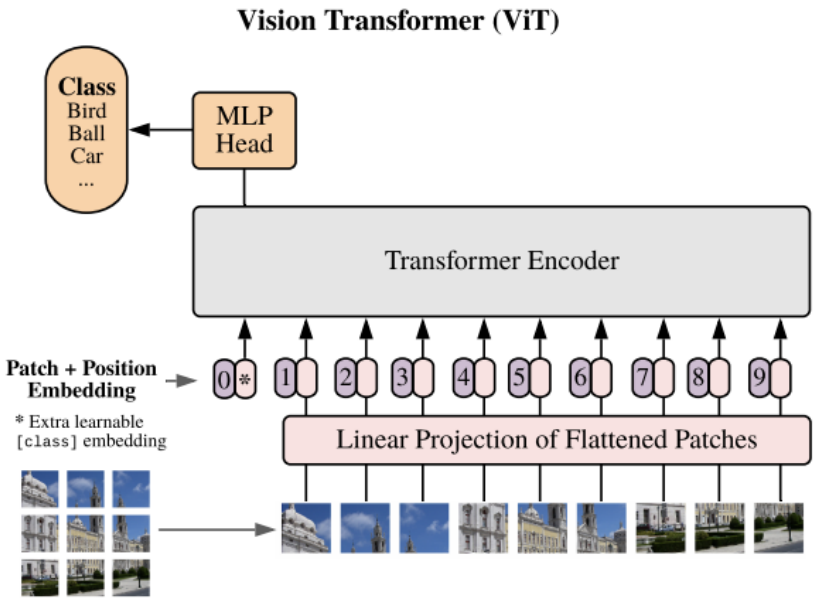


Figure 6: Overview of the Vision Transformer (ViT) workflow. The spectrogram is divided into patches, which are embedded and enriched with positional information before being processed by Transformer encoder layers. A dedicated `[class]` token collects a global representation that is passed to an MLP head for classification. The figure emphasizes the sequence-based processing of image patches that distinguishes ViTs from CNNs. *Reprinted from Dosovitskiy et al., 2021, via PapersWithCode.*

2.4.3 Linear and Interpretable Models

Alongside deep learning models, experiments with simpler and more interpretable models have also been considered in this thesis, like linear regression and shallow feedforward networks. These were mostly used when the input data was highly structured: for example, when using averaged chroma features (just 12 values per clip) or mid-level perceptual descriptors like dissonance or rhythm stability.

The benefit of these models is that they are much easier to analyze. It is possible to appreciate directly how each input feature affects the output, which makes them ideal when interpretability is a key goal. SHAP was also used in some of these experiments to better understand which features were contributing the most to the predictions.

From a mathematical point of view, a linear regression model predicts the output as a weighted sum of the input features. For a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ the prediction is

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b, \tag{4}$$

where $\mathbf{w} = (w_1, \ldots, w_p)$ are the coefficients and b is the intercept. Each coefficient indicates how strongly a feature influences the output, which makes the model easy to interpret.

To prevent overfitting, especially when the number of features grows or they are correlated, regularization is often added. In Ridge regression the objective becomes

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} \big(y_i - (\mathbf{w}^\top \mathbf{x}_i + b)\big)^2 + \lambda \|\mathbf{w}\|_2^2, \tag{5}$$

where the parameter λ controls the strength of the L_2 penalty. This discourages large coefficients and makes the solution more stable.

In some cases interaction terms between features were also added, leading to a model of the form

$$\hat{y} = \sum_{j=1}^{p} w_j x_j \; + \; \sum_{j<k} w_{jk}\, x_j x_k \; + \; b, \tag{6}$$

Here p is the number of original features, w_j are the coefficients for the individual features, and w_{jk} are the coefficients for the interaction terms, with the second summation running over all pairs of indices (j, k) such that $j < k$, which allows the model to capture how combinations of features may influence the prediction beyond their individual contributions. This extra flexibility comes at the cost of many more parameters, so regularization is crucial to keep the model from overfitting.

Even though these models did not perform as well as CNNs or ViTs, they helped me build some intuition about the relationship between musical structure and emotion, and served as a kind of benchmark to evaluate whether more complex models were really learning something meaningful or just overfitting.

2.5 Explainable AI Methods for MER

Explainable AI (XAI) encompasses a collection of techniques designed to provide insights into how machine learning models make their predictions, particularly in the case of complex, non-transparent models such as deep neural networks.

The motivations for using XAI in this project are two: first, it helps improve trust and accountability, particularly in sensitive applications; second, and perhaps more relevant in a creative domain like music, it allows us to gain a better understanding of the phenomena being modeled. In the context of MER, understanding how a model associates musical features with emotions can yield insights both for music researchers and practitioners.

This section gives an overview of the three explainability techniques used in this thesis: SHAP (SHapley Additive exPlanations), Layer-wise Relevance Propagation (LRP), and a perceptual sonification approach. These are not necessarily the most advanced or popular techniques overall, but they were selected because they fit well with the kinds of models and data used in this work (spectrograms, chromagrams, and structured features).

The goal here is to explain how these techniques work and what kind of strengths and weaknesses they have, without yet going into the details of their practical application, which will be instead covered in Chapter 4.

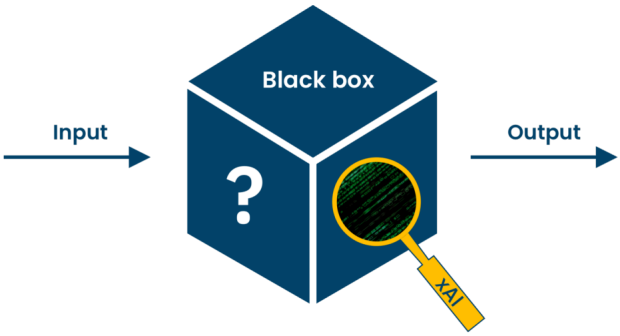


Figure 7: XAI aims to open the "black box" of a machine-learning model by making its internal decision process understandable by humans.
Reprinted from Papermaker AI, 2024.

2.5.1 SHapley Additive exPlanations (SHAP)

SHAP is based on *Shapley values*, first used in game theory to measure how much contribution each player adds to the final score. **[Lundberg and Lee, 2017]**. Here, the "game" is the prediction made by a model f, and the "players" are the input features. Let $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ be one sample with d features, and let $N = \{1, \ldots, d\}$ be the index set of all features. SHAP writes the prediction $f(x)$ as

$$f(x) = \phi_0 + \sum_{i=1}^{d} \phi_i, \tag{7}$$

where

ϕ_0 *Baseline*: the expected model output $\mathbb{E}_{z \sim D_b}[f(z)]$ over a *background set* D_b (often a random subset of the training data, but any representative distribution can be used);

ϕ_i *Shapley value* for feature i, indicating how much the specific value x_i raises or lowers the prediction relative to the baseline.

For one feature $i \in N$, the exact Shapley value sums over every subset (coalition) of the other features:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(d - |S| - 1)!}{d!} \left[f(x_{S \cup \{i\}}) - f(x_S) \right], \tag{8}$$

with

S any subset that does *not* contain i;

x_S the input where only features in S stay as they are—the others are replaced by "missing" values drawn from a background distribution;

$|S|$ the size of S; the factorial terms make sure each possible order of features is treated fairly.

The difference $f(x_{S\cup\{i\}}) - f(x_S)$ shows how much the output changes when feature i is added to coalition S. Taking the weighted average over *all* coalitions as in (8) gives a single attribution that meets the Shapley axioms listed in Table 1.

Table 1: Game-theoretic axioms underlying Shapley values and their meaning for SHAP

Axiom	**Game-theoretic requirement**	**Interpretation in the SHAP setting**
Efficiency (Completeness)	The total value produced by the grand coalition must be fully given to the players.	Baseline plus all SHAP values exactly rebuild the model output: $\phi_0 + \sum_{i=1}^{d} \phi_i = f(x)$.
Symmetry	Players that contribute in the same way to every coalition get the same reward.	Features with identical contributions across all coalitions receive the same ϕ.
Dummy (Null player)	A player that never changes the value of any coalition gets zero.	If a feature never changes the prediction ($\Delta_i(S) = 0$ for every S), then $\phi_i = 0$.
Additivity	For two games g and h combined, rewards add: $\phi_i^{g+h} = \phi_i^g + \phi_i^h$.	If we add two models (for example, by summing their outputs), the SHAP values add in the same way.

A positive ϕ_i means feature i pushes the output above the baseline ϕ_0; a negative value pushes it below. Because (7) is exact (up to the chosen approximation), adding up all ϕ_i and ϕ_0 rebuilds $f(x)$. This check helps when drawing plots like waterfall charts.

Thanks to the axioms, SHAP is *consistent*: if a feature's contribution grows after retraining, its ϕ_i cannot drop. Still, two main limitations matter for MER:

1) **Feature dependence**. Spectral bins and chroma features are often strongly correlated. Assuming full independence among features can distort the contribution that SHAP assigns to each region.

2) **Computation cost**. Estimating Shapley values on thousands of frames is computationally heavy. A practical shortcut is to aggregate statistics over musically meaningful windows (e.g., per bar) or to sample a subset of well-spaced frames.

SHAP is most useful when each feature has a clear physical or musical meaning (MFCCs, spectral centroid, rhythm features, or learned embeddings tied to the song's structure). In such cases, the additive split in (7) not only helps explain the prediction to users but also checks whether the model uses cues that make musical sense.

2.5.2 Layer-wise Relevance Propagation (LRP)

Layer-wise Relevance Propagation (LRP) is an attribution technique that redistributes a neural network's output back to the input features. **[Bach et al., 2015; Montavon et al., 2019]**.

LRP visualises the contribution of each input feature by treating the prediction value $f(x)$ as a conserved quantity and passing it backwards, layer by layer, until it reaches the inputs.

Consider a feed-forward network written as a composition of layers

$$f(x) \;=\; f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(x),$$

where $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and L is the index of the output layer. Let $a^{(\ell)}$ denote the activations in layer ℓ, $w^{(\ell)}$ the weights, and $b^{(\ell)}$ the biases. LRP assigns to every neuron j in every layer ℓ a *relevance* score $R_j^{(\ell)}$ such that

$$\sum_j R_j^{(\ell)} \;=\; \sum_k R_k^{(\ell+1)}, \qquad \ell = L-1, \ldots, 1, \tag{9}$$

and at the top layer

$$\sum_j R_j^{(L)} \;=\; f(x). \tag{10}$$

Equation (9) expresses the **relevance–conservation principle**: all relevance that leaves one layer enters the next, so that the total relevance remains exactly $f(x)$ throughout the backward sweep.

The ε-rule. For a fully connected layer, let the pre-activation of neuron k be

$$z_{jk} \;=\; a_j^{(\ell)}\, w_{jk}^{(\ell)},$$

and (optionally) $z_{bk} = b_k^{(\ell)}$ for the bias term[3]. Using the ε-rule, relevance is redis-

[3]Biases can be handled by treating each $b_k^{(\ell)}$ as the activation of an additional "bias neuron" with fixed value $a_b^{(\ell)} = 1$, so relevance is redistributed exactly as for the standard neurons. We omit z_{bk} in the main equations for clarity.

tributed as

$$R_j^{(\ell)} \;=\; \sum_k \frac{z_{jk}}{\sum_{j'} z_{j'k} \;+\; \varepsilon\,\mathrm{sign}\Big(\sum_{j'} z_{j'k}\Big)}\; R_k^{(\ell+1)}, \tag{11}$$

where a small constant $\varepsilon > 0$ (e.g. 10^{-6}) prevents division by zero and its sign matches that of the denominator. Inputs that contribute little to neuron k (small $|z_{jk}|$) receive little relevance.

Convolutional and pooling layers. For convolutional and pooling layers the same redistribution is applied locally within each receptive field, ensuring layer-wise conservation irrespective of layer type.

After the backward sweep, each input dimension x_i carries a relevance $R_i^{(0)}$ with

$$f(x) \;=\; \sum_{i=1}^{d} R_i^{(0)}.$$

A positive value $R_i^{(0)}$ supports the prediction, whereas a negative value speaks against it. Because the decomposition is *exact*, the resulting heatmap can be verified by summing all pixel (or frame) relevances and checking that the total equals the model output. These properties are usually summarized as a small set of constraints that guide how relevance is redistributed. Table 2 reports the main ones, together with their practical meaning when interpreting explanations.

Table 2: Core constraints for LRP and their practical meaning

Constraint	**Mathematical statement**	**Interpretation for explanations**
Conservation	$\sum_i R_i^{(\ell)} = \sum_k R_k^{(\ell+1)}$	All relevance credited to a layer is fully redistributed—nothing is lost or created. Ensures faithfulness at every depth.
Positivity (optional)	$R_i^{(0)} \geq 0$ under the z^+-rule	The z^+-rule discards negative evidence: only positive contributions are propagated back, so every highlighted region *supports* the prediction—often easier to interpret for non-expert users.

In summary, LRP offers a principled way to trace predictions through deep networks while exactly preserving the model output. Its relevance-conservation mirrors

SHAP's completeness axiom, making the two methods complementary: SHAP guarantees fair attribution across feature coalitions, whereas LRP guarantees layer-wise exactness with negligible extra compute.

2.5.3 Perceptual Sonification of Model Attributions

The third pillar of explainable AI explored in this thesis is *perceptual sonification*, a strategy that turns machine–generated attributions back into the same sensory modality as the original data: sound. **[Zohar et al., 2021]**. Where SHAP distributes numerical credit over input dimensions and LRP propagates relevance through the layers of a network, sonification renders those abstract scores as audible cues in the time–frequency plane, letting a listener *hear* what the model is "paying attention to." In the context of MER, the idea is both natural and novel: if the task itself concerns musical affect, the most intuitive explanation is a musical one.

From relevance to sound This subsection spells out, step by step, how a relevance heatmap $R(\tau, f)$ is converted into an audible explanation.

1. Analysis: turning the waveform into a time–frequency matrix

1.1 *Waveform.* The starting point is a continuous time signal $x(t)$, sampled at a rate F_s (samples per second) for a total duration T seconds.

1.2 *Windowing and hop size.* A short analysis window of length N_{win} samples slides over the signal in steps of H samples (the *hop size*). Each placement of the window defines a *frame index* $\tau \in \{0, \ldots, N_\tau - 1\}$, where $N_\tau = \lfloor (T\,F_s - N_{\text{win}})/H \rfloor + 1$.

1.3 *FFT size and frequency bins.* Inside each time frame we compute a N_{FFT}-point complex Fast Fourier Transform. The FFT yields $N_f = N_{\text{FFT}}$ complex coefficients, indexed by $f \in \{0, \ldots, N_f - 1\}$ and spaced F_s/N_{FFT} Hz apart.

1.4 *Short-Time Fourier Transform (STFT).* The whole procedure can be summarised as

$$X(\tau, f) \;=\; \text{STFT}\big(x\big) \in \mathbb{C}^{N_\tau \times N_f}. \tag{12}$$

where each entry $X(\tau, f)$ is a complex number that encodes both the magnitude and the phase of frequency bin f in time frame τ.

How the indices map to real units. Because τ and f are merely counters, you turn them into seconds or hertz only when you need the physical scale:

$$t_\tau = \tau \frac{H}{F_s} \text{ [s]}, \qquad f_f = f \frac{F_s}{N_{\text{FFT}}} \text{ [Hz]}. \tag{13}$$

Thus the hop size H fixes the time resolution, while the FFT size N_{FFT} fixes the frequency resolution.

2. Attribution: where the model looks

The application of *Layer-wise Relevance Propagation* on the Vision Transformer returns a non-negative matrix

$$R(\tau, f) \in \mathbb{R}_{\geq 0}^{N_\tau \times N_f}, \tag{14}$$

whose entry $R(\tau, f)$ quantifies *how much the model relies on the information in time frame τ and frequency bin f.* Classical explainability stops here, showing R as a coloured heatmap. We go one step further and render R in the sonic domain.

3. Relevance-weighted spectrum

We amplify the magnitude of each STFT coefficient in proportion to its relevance, keeping the phase untouched so that temporal micro-structure is preserved:

$$\widehat{X}(\tau, f) \;=\; A\big(R(\tau, f)\big)\, X(\tau, f), \tag{15}$$

where $A(\cdot)$ is any monotonically increasing *gain function.*

4. Re-synthesis

Applying the inverse STFT to the relevance-weighted spectrum produces a new waveform $\tilde{x}(t)$:

$$\tilde{x}(t) \;=\; \mathrm{iSTFT}\big(\widehat{X}\big), \tag{16}$$

which is almost identical to the original $x(t)$, *except* that time–frequency regions the model considered important now sound noticeably louder. The listener literally hears the parts the network "attends to," offering an intuitive, audio-native explanation.

Fixed numerical settings—window length, hop size H, FFT size N_{FFT}, frame-smoothing of $R(\tau, f)$, and the particular (γ, p) chosen for A—are listed in Section 4. The four numbered steps above constitute the general pipeline, independent of those implementation details.

3 Related Work

3.1 Traditional and Deep Learning Approaches to MER

Earlier attempts at MER mostly relied on classic machine learning models like Support Vector Machines (SVMs) or decision trees. These models used hand-crafted features such as Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid, or energy, extracted through standard signal processing techniques. Even though this approach was reasonably effective in some cases, it often failed to capture the deeper structure of music and the subtle ways in which it relates to emotions.

In recent years, particularly since 2018, deep learning has become the dominant method in MER. These models do not rely on pre-defined features but instead learn directly from data, often using audio representations like spectrograms or mel-spectrograms. CNNs have proven especially useful for capturing spatial patterns in time-frequency representations, while Recurrent Neural Networks (RNNs) have been used to model temporal dependencies in music. For instance, **[Chowdhury et al., 2019]** demonstrated that even relatively simple CNN architectures could learn to predict emotional values directly from log-mel spectrograms.

The release of the PMEmo dataset in 2018 **[K. Zhang et al., 2018]** also marked a key step forward. It provided real-world songs annotated with continuous valence and arousal labels, which helped standardize training and evaluation procedures **[Y. Zhang et al., 2018]**. However, despite these advances, most deep models still operate as black boxes. They might perform well, but it is hard to understand why a certain output is produced, which is a limitation especially in research and creative applications.

Some studies have combined multiple time–frequency representations to improve MER accuracy. For example, in **[Er and Aydilek, 2019]**, chroma features, which capture harmonic content, are fused with spectrograms, which encode spectral–temporal patterns. This multi-representation approach outperforms using either feature set alone, showing their complementarity. Although it does not address explainability, this work is closely related to part of the methodology developed later in this thesis, where chroma and spectrogram information are also used together, making it a

relevant point of reference.

3.2 Use of Mid-level Features in MER

One of the most promising directions for making MER more interpretable has been the use of mid-level features: descriptors that sit somewhere between raw audio data and emotional judgments. These include things like rhythmic stability, melodic complexity, and dissonance, which are more intuitive for humans to understand.

In their 2019 work, **Chowdhury et al., 2019** proposed a two-step model where audio is first mapped to seven mid-level features, which are then used to predict emotional scores through a simple regression layer. This design makes it possible to explain predictions in terms of perceptual qualities of the music, rather than low-level spectral features.

Building on this idea, the current thesis explores whether using even simpler or more symbolic features, like chords or average chroma, can improve interpretability without drastically reducing performance. The hope is that such features might be more actionable for musicians or analysts who want to understand or control the emotional tone of a piece.

The mid-level features used in the original model were derived from perceptual studies like those by Friberg et al. and Aljanaki et al. **[Aljanaki and Soleymani, 2018b; Friberg et al., 2014]**, and provide a helpful bridge between psychological theories of emotion and audio-based models.

3.3 Mid-level Bottleneck Architectures

Chowdhury et al. (2019) ask whether a deep model for music–emotion recognition (MER) can remain explainable if its predictions must pass through a small set of perceptually meaningful intermediate cues. Using the 360-excerpt *Soundtracks* corpus annotated for both emotions (Table 3) and the seven mid-level perceptual features (Table 4) of **Aljanaki and Soleymani (2018a)**, **Eerola and Vuoskoski (2011)** compare three CNN-based architectures (see Figure 8). Predictive accuracy

is measured as the Pearson correlation between predicted and reference emotion ratings; the "cost of explainability" is the loss incurred when the mid-level bottleneck is enforced.

Table 3: Emotion categories and dimensions provided with the 360-excerpt *Soundtracks (Stimulus Set 1)* corpus **[Eerola and Vuoskoski, 2011]**. Expert raters scored each excerpt on a 1–7.83 Likert scale (later multiplied by 0.1 in our experiments).

Emotion label	**Meaning / rating prompt**
Happy	Degree to which the excerpt expresses a cheerful, positive mood.
Sad	Degree of sadness, melancholy or grief conveyed.
Tender	Perceived warmth, softness or loving tenderness.
Fearful	Level of fear, anxiety or suspense communicated.
Angry	Strength of anger, aggression or hostility expressed.
Valence	Position on the positive–negative affect axis (high = pleasant, low = unpleasant).
Energy	Perceived activity or intensity (low = calm/relaxed, high = vigorous/energetic).
Tension	Psychological stress or suspense experienced (low = relaxed, high = tense).

To explore the trade-off between explainability and predictive performance, the authors design and compare three neural architectures. Each of them reflects a different level of constraint on the information flow from audio to emotion prediction, as summarised below.

A2E. This baseline maps a log-mel spectrogram directly to the continuous emotion ratings using a convolutional network followed by a small fully connected block. Because the prediction bypasses any explicit musical descriptors, A2E delivers the highest raw accuracy but offers no insight into *why* a particular score is produced. The metrics scores obtained with this architecture will work as reference when we will compare the ones that pass through the mid-level features to compute the "cost of explainability".

A2Mid2E. To add interpretability, A2Mid2E routes the same audio network through the seven mid-level perceptual features listed in Table 4 (melodiousness, articulation, rhythm stability, rhythm complexity, dissonance, tonal stability and "minorness"). The model first learns to reproduce these descriptors and then combines them into the final emotion estimates. Because every prediction passes through this seven-value bottleneck, users can inspect which perceptual cues drove a given emotional

assessment.

A2Mid2E-joint. The joint variant retains the bottleneck but optimises descriptor accuracy and emotion accuracy simultaneously. Allowing the whole network to adapt to the downstream task recovers most of the performance lost in A2Mid2E, while still preserving a clear, seven-dimensional explanation layer. The difference in correlation between A2Mid2E-joint and the direct A2E baseline is therefore interpreted as the *cost of explainability*.

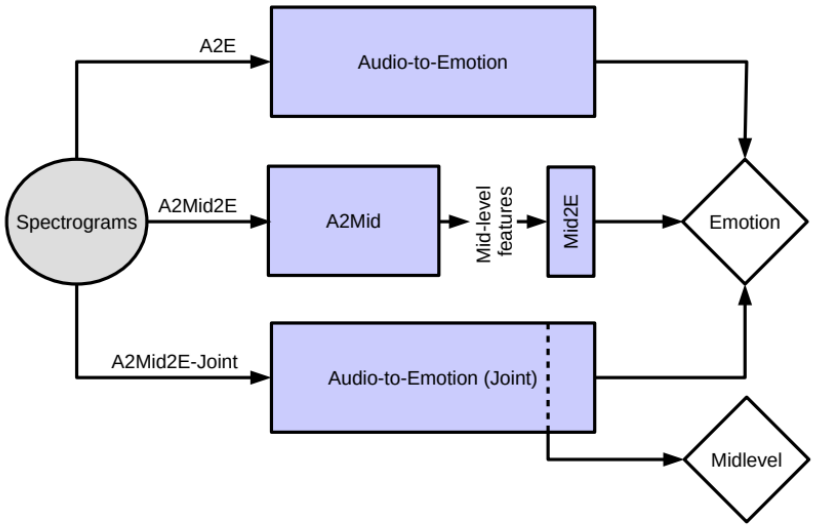


Figure 8: The three architectures compared for predicting emotion from audio (A2Mid, A2Mid2E and A2Mid2E-joint). *Reprinted from Chowdhury et al., 2019.*

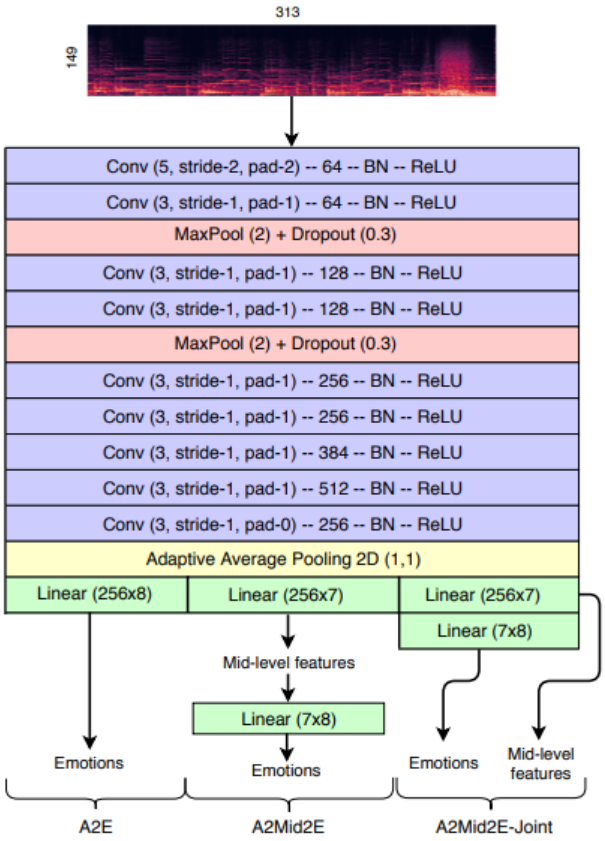


Figure 9: Visual representation of the convolutional backbone shared by all networks up to the *Adaptive Average Pooling 2D* layer. *Reprinted from Chowdhury et al., 2019.*

All three variants in Figure 8 rely on the same convolutional backbone shown in Figure 9. This design is directly taken from **Chowdhury et al., 2019** and follows a fairly standard VGG-style structure. The use of 3×3 convolutions with progressively increasing channel depth (from 64 up to 512) is motivated by the idea of capturing simple local patterns at the beginning, such as spectral edges or harmonic stacks, and gradually combining them into richer, more abstract representations. Batch normalization and ReLU are included after each convolution to stabilize training and introduce non-linearity, while max-pooling reduces the spatial dimensions and introduces invariance to small shifts in time or frequency. Dropout (set to 0.3) is used as a regularizer to prevent overfitting, which is particularly important given the relatively small size of the Soundtracks dataset. Finally, the adaptive average pooling layer reduces each feature map to a fixed 1×1 representation, making the output size independent of the input dimensions and suitable for the fully connected layers that follow.

In other words, the backbone is not meant to be a novel proposal, but rather a

controlled and reproducible choice: it is complex enough to learn useful spectro-temporal features, but at the same time lightweight compared to more recent deep architectures, which makes it well suited for testing the effect of the mid-level bottleneck without adding unnecessary confounding factors.

Table 4: Overview of the seven mid-level perceptual features that form the bottleneck in **Chowdhury et al., 2019**. Each descriptor is accompanied by the prompt used in **Aljanaki and Soleymani, 2018a**. Following **Chowdhury et al., 2019**, the feature *Modality* is referred to as *"Minorness"*.

Perceptual Feature	**Question asked to human raters**
Melodiousness	To which excerpt do you feel like singing along?
Articulation	Which has more sounds with staccato articulation?
Rhythmic Stability	Imagine marching along with the music. Which is easier to march along with?
Rhythmic Complexity	Is it difficult to repeat by tapping? Is it difficult to find the meter? Does the rhythm have many layers?
Dissonance	Which excerpt has noisier timbre? Has more dissonant intervals (tritones, seconds, etc.)?
Tonal Stability	Where is it easier to determine the tonic and key? In which excerpt are there more modulations?
Modality ("Minorness")	Imagine accompanying this song with chords. Which song would have more minor chords?

3.4 Sonification and Perceptual Explanations in Music AI

While most XAI techniques rely on visual outputs like plots or heatmaps, some researchers have started exploring whether model explanations can be conveyed through sound: a direction known as sonification.

One example is **AudioLIME**, an adaptation of the LIME framework for audio data. It works by segmenting the audio input, evaluating the contribution of each segment to the prediction, and then resynthesizing the audio to make the relevant parts louder or clearer **[Ribeiro et al., 2016; Zohar et al., 2021]**.

The idea is to turn the explanation into something we can hear, rather than just see. If a segment classified as "Happy" emphasizes melodic or rhythmic peaks, the listener might intuitively grasp what the model is focusing on. Although still experimental,

this kind of perceptual feedback has potential both for user interaction and for validating model behavior in a more intuitive way.

To give some more context, LIME itself **[Ribeiro et al., 2016]** is a model-agnostic method that perturbs the input and then trains a simple surrogate model to approximate the decision boundary of the complex model around a given example. In AudioLIME this principle is applied to sound: the input is divided into chunks of audio, these chunks are turned on or off in different combinations, and the effect on the prediction is measured. The explanation can then be reconstructed not only visually but also through listening, by reassembling the audio and making the influential parts more prominent. This makes the abstract idea of "feature importance" more concrete for the listener.

Beyond AudioLIME, the idea of using sonification for explanations has been considered in a broader sense in the literature. Surveys like the ones from **Grond and Berger, 2011** and **Dubus and Bresin, 2013** have collected many examples of how data can be mapped to sound in scientific and perceptual studies, arguing that listening can sometimes reveal patterns that are hard to notice visually. Other more recent works such as **Gresham et al., 2020** discuss auditory displays specifically for machine learning explanations, suggesting that hearing what the model pays attention to might give users a more natural sense of trust and understanding.

In this thesis, a similar approach is explored using a Vision Transformer trained on spectrograms, combined with LRP for attribution. The resulting relevance maps are used to modify the volume of the original audio: louder segments correspond to parts the model found more important for emotion classification. This is applied to clips from the "Musical Emotions Classification" dataset available on Kaggle **[“Musical Emotions Classification”, 2020]**.

Our implementation is directly based on the ViT-LRP framework proposed by **[Chefer et al., 2021]**. They fine-tuned a ViT-Base/16 model **[Dosovitskiy et al., 2021]**, where the input spectrogram was resized to 224×224 pixels, divided into 16×16 patches that are treated as tokens and processed through 12 transformer layers with 12 attention heads (embedding dimension 768). The resulting relevance maps obtained through Chefer's method were then used as input for the sonification

stage.

Overall, the literature suggests that combining structured features with perceptual tools might be a promising direction for making MER models not only more accurate but also more transparent.

4 Methodology

This chapter presents how the two contributions of the thesis — the one based on musical features and the one based on perceptual explanations — were practically implemented. The goal here is not just to describe the experiments, but also to give a clear idea of how everything was built step by step, from data preparation to model training, explainability and sonification.

4.1 Research Design

The experimental design of this thesis is structured around two distinct but complementary approaches to understanding how machine learning models perceive and predict musical emotion. Both approaches rely on supervised learning, but they differ significantly in the way they treat the input data and the kind of explanations they can provide.

The first approach focuses on building models using a variety of audio-derived features, which are representations that are extracted or computed from the audio signal itself. These include low and mid-level descriptors such as spectrograms, chromagrams, average chroma vectors, mid-level perceptual features (like melodiousness, articulation, rhythm complexity), and even symbolic approximations of chord sequences. Although these features differ in dimensionality and abstraction, they share a common trait: they are all derived from the audio and can be related, more or less directly, to musical concepts. The goal in this part of the work is to build interpretable models that connect changes in these features with variations in emotional perception. In some cases, we also apply post-hoc explanation tools, such as SHAP, to better visualize which features influenced the model's outputs the most.

The second approach takes a more perceptual and model-centric perspective. Instead of trying to interpret the input features, it focuses on explaining what the model has learned when trained directly on spectrograms. Specifically, we use a ViT architecture and apply LRP to produce heatmaps that highlight the regions of the input spectrogram most responsible for a given decision. These heatmaps are

then used to sonify the model's attention, effectively producing a modified version of the original audio that emphasizes the parts deemed most emotionally relevant by the model itself. This method shifts the focus of interpretability from structured inputs to perceptual salience, offering a way to both visualize and hear what the model "thinks" is important.

While these two directions are different in spirit (one rooted in music theory and structured features, the other in perceptual relevance), they were both designed to investigate the same question: how can we build emotion recognition systems that are not only accurate, but also explainable in human terms? This thesis progresses by alternating between these two perspectives, presenting experiments that highlight their respective strengths, limitations, and points of contact.

The concrete model architectures employed in each route have already been introduced in Chapter 3. For the feature-based approach, we rely on the CNN and mid-level bottleneck architectures described by **Chowdhury et al., 2019**, while for the perceptual route we fine-tune a ViT-Base/16 with LRP as in **Chefer et al., 2021**. In this chapter, we therefore focus on the data preparation, training protocols and evaluation procedures, rather than re-describing the network structures.

In both cases, we followed a supervised learning approach, meaning that the models were trained on input–output pairs where the target emotion labels were known in advance. Depending on the dataset and experiment, the task was either regression (predicting continuous emotion scores such as valence or arousal) or classification (e.g., distinguishing between Happy and Sad). Most of the experiments were run in Google Colab, taking advantage of GPU acceleration to reduce training time and facilitate experimentation.

4.2 Datasets Used

For our first experiments, we leverage music recordings annotated both with mid-level perceptual features, and with human ratings along some well-defined emotion categories. Our starting point is Aljanaki & Soleymani's *Mid-level Perceptual Features* dataset **[Aljanaki and Soleymani, 2018a]**, which provides mid-level feature annotations. For the actual emotion prediction experiments, we then use the *Sound-*

tracks dataset **[Eerola and Vuoskoski, 2011]**, which is contained in the Aljanaki collection as a subset, and comes with numeric emotion ratings along 8 dimensions.

In addition to this, the *Musical Emotion Classification* dataset from Kaggle **["Musical Emotions Classification", 2020]** is the dataset that has been used for the sonification experiments, allowing research in the classification task in the previously descripted perceptual and model-centric perspective.

These datasets are presented in the next sections, and the summary of their characteristics is presented in Table 5

4.2.1 Mid-Level Feature Dataset

The Mid-level Perceptual Features Dataset **[Aljanaki and Soleymani, 2018a]** consists of 5000 song snippets of around 15 seconds each annotated according to the seven mid-level descriptors listed in Table 4. The annotators were required to have some musical education and were selected based on passing a musical test. The ratings range from 1 to 10 and were scaled by a factor of 0.1 before being used for the experiments.

4.2.2 Soundtracks Dataset

The Soundtracks (Stimulus Set 1) dataset **[Eerola and Vuoskoski, 2011]** consists of 360 excerpts from 110 movie soundtracks. The excerpts come with expert ratings for five categories following the discrete emotion model (happy, sad, tender, fearful, angry) and three categories following the dimensional model (valence, energy, tension). This makes it a suitable dataset for musically conveyed emotions. The ratings in the dataset range from 1 to 7.83 and were scaled by a factor of 0.1 before being used for our experiments. As stated above, all the songs in this set are also contained in the Mid-level Features Dataset, so that both kinds of ground truth are available.

4.2.3 Musical Emotions Classification (Kaggle)

The Musical Emotions Classification dataset includes about 2000 audio samples, each 10 seconds long, labeled as either "Happy" or "Sad." Most of the original sources of the audios were longer than 10 seconds and generated more than one 10-second-long clip for the dataset. These clips were turned into spectrograms and used to train the Vision Transformer (ViT) model and to test attribution methods like LRP and the sonification pipeline. The dataset is publicly available on Kaggle **["Musical Emotions Classification", 2020]**.

Table 5: Key statistics of the music–emotion datasets used in this thesis.[4]

Characteristic	Mid-level Features	Soundtracks	Musical Emotions
Number of clips	5 000	360 (110 tracks)	2 126
Clip length	15 s	15 s	10 s
Label type	7 mid-level	5 discrete + 3 dim.	Binary (Happy/Sad)
No. of attributes	7	8	1
Label scale	1–10 (scaled $\times 0.1$)	1–7.83 (scaled $\times 0.1$)	categorical

4.3 Feature Extraction Techniques

Depending on the experiment, we used different representations of the audio, including spectrograms, chroma features, and symbolic data like chords.

4.3.1 Spectrograms

Spectrograms (see Section 2.3.1) were one of the primary representations used throughout this work. Although they originate from a well-established audio analysis technique, in this context they were treated primarily as visual inputs for deep learning models. By converting audio signals into two-dimensional time–frequency representations, it becomes possible to take advantage of computer vision architectures to analyze musical structure and emotional content.

The rationale behind using spectrograms lies in their capacity to capture fine-grained temporal and spectral features, including harmonic patterns, rhythmic cues, and

[4] All audio excerpts are down-mixed to mono and resampled to 22 050 Hz for the experiments reported in Chapters X–Y.

dynamic contours, which are all known to correlate with perceived emotion. Once generated, each spectrogram was saved as an image and used directly as input to the models, either as a grayscale or RGB representation depending on the architecture.

From a methodological point of view, this choice implies a shift from treating music as a sequence of notes or features to treating it as a surface of energy across time and frequency. This allowed us to explore a different set of architectures (CNNs and ViTs) and to apply visual explainability techniques, such as heatmaps and relevance propagation, which would not have been possible using only symbolic or low-dimensional representations.

The exact parameters used for spectrogram generation, such as STFT configuration, frequency scaling, and image resizing, are reported in Chapter 5, where they are discussed in the context of specific experiments and models.

Spectrograms were used for both approaches of the thesis (with different dimensions to satisfy the requirements of the different tasks) as they are one of the most complete and interesting representations of audio. In the first part of experiments they were used as inputs to learn the mid-level features while in the second approach they were the data on which the ViT model with LRP worked to obtain the relevance maps that were used for sonification. In particular, for the lattest (sonification experiments), spectrograms were the only audio representation data used.

4.3.2 Chromagrams

Chromagrams (see Section 2.3.2) represent the distribution of spectral energy across the twelve pitch classes of the musical scale, regardless of the octave. This representation abstracts away information about timbre and precise pitch, focusing instead on harmonic content and tonal structure. Because of this, chroma features are particularly suited for tasks involving chord recognition, key detection, and, in our case, emotion prediction based on harmonic patterns.

In this work, chroma features were extracted from audio using standard signal processing techniques and were treated as compact, structured inputs for machine learning models. Their relatively low dimensionality (typically 12 bins per frame) made

them ideal for use in models that prioritize interpretability, such as linear regressors or shallow neural networks.

Unlike spectrograms, chromagrams emphasize the musical dimension rather than the acoustic dimension of the signal. They allowed us to investigate to what extent harmonic cues alone can be predictive of emotion, without relying on detailed timbral or rhythmic information. In particular, we experimented both with raw chroma sequences and with aggregate statistics such as the average chroma profile across each segment.

The technical details regarding chroma extraction (e.g. windowing, hop length, normalization) are discussed later in Chapter 5 alongside the specific models that employed them.

Chromagrams were used just for the first approach to derive more audio-related features.

4.4 Chord Features

In an effort to enrich the mid-level feature set with harmonic content, a pipeline to extract chords directly from audio using the `Essentia` library was explored. Specifically, the `ChordsDetection` algorithm was applied on 10-second excerpts from a subset of 360 annotated audio files. The process involved computing Harmonic Pitch Class Profiles (HPCP) over overlapping frames and then aggregating the pitch profiles to detect the most likely chord labels.

These vectors were concatenated with the previously presented perceptual descriptors, forming the input to a feed-forward neural network trained to predict emotion scores. To see how these extracted chords influenced the decision of the machine learning model, SHAP was applied.

4.5 Adapted Architectures

Previously, in Section 2.4, we provided an overview of the main model families used in Music Emotion Recognition, and in Chapter 3 we discussed the specific baselines

proposed in prior work. In this section, the focus shifts to the practical adaptations applied in our implementation. The aim is not to reintroduce the architectures themselves, but to clarify how they were modified to accommodate the different feature sets explored in this thesis.

A2E (Audio → Emotion)

We follow the convolutional backbone of **Chowdhury et al., 2019**, keeping the architecture unchanged but varying the input representation.

Spectrograms. Spectrograms are fed as 2D time–frequency inputs with a singleton channel ($H \times W \times 1$), e.g. $(149, 313, 1)$. The CNN architecture remains identical to the baseline, with input and output dimensions identical to the ones described in Section 2.4.

Chromagrams. Chromagrams have shape $(12, 431, 1)$. Again, the CNN backbone is unchanged: the convolutions simply operate over a smaller, pitch–class domain instead of a time–frequency surface. Therefore the architecture is the same and has the same output dimension, but the input dimensions are changed in order to work with the different image.

Averaged chroma (C2E). When chroma is averaged over time, the result is a 12-dimensional vector. Since convolution presupposes spatial locality, which a flat vector lacks, a CNN would add parameters without exploiting any meaningful inductive bias. For this reason, we replaced the CNN with a multilayer perceptron (MLP) composed of stacked Dense, BatchNorm, and Dropout layers. Conceptually, both CNNs and MLPs learn hierarchical transformations, but while CNNs exploit local 2D patterns, MLPs are more suitable for tabular inputs where features interact globally. This adaptation drastically reduces complexity and risk of overfitting.

Multi-input (spectrogram + chromagram). In addition to the single-input variants, we designed a *multi-branch* version of A2E. In this case, the spectrogram

$(149, 313, 1)$ and the chromagram $(12, 431, 1)$ are each fed into a separate convolutional backbone, structurally identical to the one used in the single-input setting but with independent weights. This means that the two feature types are processed in parallel streams: one CNN learns filters specialized for time–frequency patterns (timbral and rhythmic cues), while the other learns filters specialized for pitch-class sequences (harmonic cues).

At the end of each branch, the high-level embeddings are reduced to fixed-size vectors through global average pooling and flattening. These two embeddings, one per input modality, are then concatenated into a joint representation of dimension 512. Only at this point do the two information streams interact: the concatenated vector is passed through a shared dense head that outputs the final emotion prediction.

This design corresponds to a *late fusion* strategy: each modality is first processed independently, preserving the same architecture used before, and only afterwards are the learned representations combined, producing a combined output. The rationale is to let the network specialize on complementary aspects of the signal before merging them into a unified decision.

A2Mid2E (Audio → Mid-level → Emotion)

Here the CNN backbone again follows **Chowdhury et al., 2019**, trained to predict the seven mid-level perceptual descriptors. Our adaptation concerns the bottleneck layer: to the original seven features we concatenated additional harmonic descriptors (12-d average chroma and 24 chord flags), producing bottlenecks of size 7, 19, and 34. These numbers of features will be explained more in detail lateer. The downstream regressor is unchanged; only its input dimensionality grows.

Vision Transformer with LRP (Perceptual Route)

For the perceptual route, we directly build on the work of **Chefer et al., 2021**, who proposed an extension of the Vision Transformer with Layer-wise Relevance Propagation (LRP) and attention rollout for interpretability. We reused their publicly available implementation, fine-tuning the ViT on spectrogram images from our

dataset. The model was then used to produce class-specific relevance maps, which were subsequently employed in the sonification stage (Section 5.2). In practice, this means that our perceptual experiments follow the same architecture and explanation pipeline described by **Chefer et al., 2021**, with minimal changes other than adapting the input format and training procedure to our task.

4.6 Evaluation Metrics

For the first set of experiments related to the mid-level features, to ensure comparability with **Chowdhury et al., 2019** we adopt two complementary metrics: Pearson's correlation coefficient (PC) and Mean-Squared Error (MSE). For the sonification route, in which the task was no longer regression but switched to the prediction of the class "Happy" or "Sad" we used the classic Cross-Entropy (CE) metric as loss for the fine-tuning of the ViT model with the spectrograms, while the accuracy was used as evaluation metric. Finally, precision, recall and the F1 score formulas are presented as they will be considered as indicators for the class predictions when discussing the confusion matrix in section 5.2.2.

Pearson's correlation coefficient. PC measures the strength and direction of the linear relationship between the ground-truth values $\{y_i\}_{i=1}^n$ and the model predictions $\{\hat{y}_i\}_{i=1}^n$:

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})\,(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\,\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{17}$$

where $\bar{y}$ and $\bar{\hat{y}}$ denote the sample means of the true and predicted values, respectively. A coefficient close to $+1$ or -1 indicates a strong positive or negative linear association, whereas values near 0 imply little or no linear correlation.

Mean-Squared Error. MSE is used as loss function, meaning it is the metric that we minimize during model training, and it appears in the reported results to

complement PC with an absolute accuracy measure:

$$\text{MSE} \;=\; \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{18}$$

Lower MSE values indicate smaller average squared deviations between predictions and targets, hence better performance.

Cross-Entropy Loss. CE is the standard loss for multi-class classification. For each instance i with true one-hot label vector $\mathbf{y}_i = (y_{i1}, \ldots, y_{iK})$ and predicted class probabilities $\hat{\mathbf{p}}_i = (\hat{p}_{i1}, \ldots, \hat{p}_{iK})$ over K classes, the (categorical) cross-entropy is

$$\text{CE} \;=\; -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\,\log \hat{p}_{ik}. \tag{19}$$

The loss penalises confident but wrong predictions heavily, encouraging the model to assign high probability to the correct class and low probability to the others. Lower CE values therefore correspond to better class-prediction accuracy.

Accuracy. Accuracy is the most straightforward and widely used evaluation metric in classification tasks. It is defined as the ratio between the number of correctly predicted instances and the total number of instances:

$$\text{Accuracy} \;=\; \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \tag{20}$$

A higher accuracy indicates that a larger fraction of samples were classified correctly. While it provides an intuitive global measure of performance, accuracy may be less informative in scenarios with class imbalance, since it does not account for the distribution of errors across classes.

Precision. Precision measures the proportion of instances that were predicted as positive and are actually positive. It is defined as

$$\text{Precision} \;=\; \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \tag{21}$$

High precision indicates that the model makes few false positive errors.

Recall. Recall, also called sensitivity or true positive rate, quantifies the ability of the model to identify all relevant instances of the positive class:

$$\text{Recall} \;=\; \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \tag{22}$$

High recall means that most of the actual positives are correctly detected.

F1 Score. The F1 score is the harmonic mean of precision and recall, providing a single measure that balances the two:

$$\text{F1} \;=\; 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{23}$$

It is particularly useful when the dataset is imbalanced, as it penalises models that achieve high precision but low recall, or vice versa.

5 Experiments and Discussion

This section pulls together the results from the two parts of the thesis and tries to make sense of the numbers interpreting them. The common thread is explainability: the goal was never "highest score at all costs", but models that we can read, question, and reuse as tools. In practice, this meant (i) using a mid-level bottleneck with simple heads where coefficients and feature roles are visible, and (ii) turning attributions into sound so we can literally hear what the model relied on.

Across both routes a few themes recur. Harmonic and timbral cues seem complementary, not interchangeable. Simpler models are often more stable when features are correlated. And attribution, whether plotted or sonified, helps to sanity-check behaviour: when the model is right, the highlighted/boosted regions usually make musical sense; when it is wrong, the maps and audio often show why. Of course there are limits (dataset size, down/up-sampling to fit the ViT, simple chord encodings), so the discussion keeps claims modest and points to where more data or different encodings would likely help.

The rest of this section is split in two parts. First, I discuss the mid-level feature experiments (A2E and A2Mid→Mid2E): what worked, what did not, and what the importance analyses suggest. Then we move to the sonification route (ViT + LRP), looking at the confusion matrix, the four qualitative examples, and what the sonified attributions tell us about the model's habits. This section presents the experiments conducted on the two parts of the thesis: mid-level features attribution on first and then sonification.

5.1 Experiments on mid-level musical features

This section studies how the model's understanding of music shifts when we change the features and the model's structure. The three model configurations used for these experiments are:

1) **A2E** — a single model maps the raw audio directly to emotion ratings;

2) **A2Mid** — the model first predicts seven interpretable mid-level musical descriptors;
3) **Mid2E** — those descriptors, optionally extended with chroma and chord information, are fed into a second model that outputs the final emotion scores.

The main goal is *explainability*. We track how regression coefficients and SHAP attributions shift when we

- add or remove feature sets (spectrogram, chroma, chords, mid-level descriptors), and
- switch between linear regressors and shallow neural networks.

These comparisons reveal which musical cues gain or lose influence and show how each modelling choice shapes the system's internal reasoning about emotion. Performance is reported as mean ± standard deviation under the two metrics introduced in subsection 4.6:

- Pearson's correlation coefficient (r) for directional agreement;
- Mean-Squared Error (MSE) for absolute deviation;

Looking at the A2E and A2Mid→Mid2E results, two things will stand out. First, when we predict emotion directly from audio, putting chromagram and spectrogram together works best (Table 10). This makes sense: chroma brings harmonic context, the spectrogram brings timbre and broadband energy, and the model seems to use both. Second, in the bottleneck setup (A2Mid→Mid2E), the choice of model matters more than stacking lots of related features. The linear head behaves well and is stable, while the shallow neural network tends to overfit (Table 12, Table 14).

5.1.1 A2E – Audio to Emotion

The first block of experiments establishes four baselines that feed the convolutional network of Figure 9 with different audio representations. Each model is trained on the *Soundtracks* dataset and evaluated on the held-out test set which was set to contain 20% of the data, but different reppresentations of the audio file are being used as input to the model each time. In order to provide a more precise result

with standard deviation included, each experiment was run four times with different random seeds.

Spectrogram input Table 6 shows that a plain log-magnitude spectrogram is a strong baseline. The model reaches a mean global correlation of 0.688 ± 0.060 and an MSE of 2.180 ± 0.300. Energy is captured best ($r = 0.775$), while "Happy" and "Sad" remain harder to predict, hinting that the frequency information contained in the spectrogram alone carries limited information for those two emotions.

Table 6: A2E performance with *log-magnitude spectrogram* input.

Metric	**Mean ± SD**
Total correlation	0.688 ± 0.060
Total MSE	2.180 ± 0.300
Per-emotion correlations	
Valence	0.679 ± 0.072
Energy	0.775 ± 0.051
Tension	0.684 ± 0.056
Anger	0.710 ± 0.076
Fear	0.665 ± 0.060
Happy	0.460 ± 0.120
Sad	0.440 ± 0.130
Tender	0.619 ± 0.067

Chromagram input

With pure harmonic information (Table 7) the overall numbers stay close to the spectrogram case ($r = 0.682$, MSE $= 2.240$). Valence even rises slightly, but Energy drops, suggesting that chroma is better at capturing pleasantness than intensity. Variation across random seeds is also lower, pointing to a more stable training process.

Table 7: A2E performance with *chromagram* input.

Metric	**Mean ± SD**
Total correlation	0.682 ± 0.012
Total MSE	2.240 ± 0.140
Per-emotion correlations	
Valence	0.706 ± 0.021
Energy	0.636 ± 0.060
Tension	0.688 ± 0.017
Anger	0.613 ± 0.067
Fear	0.695 ± 0.027
Happy	0.503 ± 0.062
Sad	0.487 ± 0.034
Tender	0.590 ± 0.044

Combined chromagram + spectrogram

Combining chromogram and spectrogram (Table 8) gives the best results so far: $r = 0.730$ and MSE $= 1.990$. Gains are consistent across all emotions, with Valence and Happy benefiting the most. The drop in MSE suggests that the two feature sets carry complementary information rather than redundant data.

Table 8: A2E performance with concatenated *chromagram + spectrogram* input.

Metric	**Mean ± SD**
Total correlation	0.730 ± 0.022
Total MSE	1.990 ± 0.120
Per-emotion correlations	
Valence	0.772 ± 0.048
Energy	0.745 ± 0.025
Tension	0.748 ± 0.024
Anger	0.715 ± 0.068
Fear	0.721 ± 0.047
Happy	0.603 ± 0.023
Sad	0.516 ± 0.037
Tender	0.668 ± 0.042

Simplified 12-bin chromagram

Aggregating the chroma into 12 coarse bins (Table 9) hurts performance badly: correlation drops to 0.518 and MSE almost doubles to 3.790. Fine-grained spectral detail seems essential, and also their change over time; over-simplifying the feature space throws away too much information.

Table 9: A2E performance with *mean 12-bin chroma* input.

Metric	**Mean ± SD**
Total correlation	0.518 ± 0.042
Total MSE	3.790 ± 0.270
Per-emotion correlations	
Valence	0.508 ± 0.051
Energy	0.366 ± 0.050
Tension	0.520 ± 0.120
Anger	0.393 ± 0.094
Fear	0.539 ± 0.042
Happy	0.338 ± 0.082
Sad	0.321 ± 0.056
Tender	0.400 ± 0.140

Table 10: Direct comparison of all A2E variants.

Input	**Total corr.**	**Total MSE**
Spectrogram	0.688 ± 0.060	2.180 ± 0.300
Chromagram	0.682 ± 0.012	2.240 ± 0.140
Chroma + Spectrogram	0.730 ± 0.022	1.99 ± 0.12
12-bin mean chroma	0.518 ± 0.042	3.790 ± 0.270

Summary of A2E configurations

Table 10 shows what seems to be the solution: mixing harmonic (chroma) and timbral (spectrogram) cues gives the largest boost, while over-simplified chroma features drag the model down. The results confirm that emotion perception in music depends on both pitch structure and detailed spectral texture, and that keeping a richer feature set helps the network capture this blend.

The concatenated input consistently beats either source alone (Table 8). Roughly speaking, valence-like behaviour seems tied to pitch organisation (chroma), while energy/tension are captured more from patterns that the spectrogram captures.

When we collapse harmony to a time-averaged 12-bin chroma (no temporal evolution), performance drops a lot (Table 9). That suggests we do need both finer spectral detail and how it changes over time.

5.1.2 A2Mid – Audio to Mid-Level Features

Predicting the seven perceptual descriptors of Table 4 from audio is the critical "bottleneck" step in the explainable pipeline. These experiments were repeated six times instead of four to reduce variance, because accurate predictions of these mid-level features is important.

Table 11: Prediction accuracy for each mid-level feature (A2Mid stage).

Metric	**Mean ± SD**
Total correlation	0.624 ± 0.035
Total MSE	0.026 ± 0.002
Per-feature correlations	
Melodiousness	0.620 ± 0.070
Articulation	0.841 ± 0.041
Rhythm complexity	0.332 ± 0.081
Rhythm stability	0.300 ± 0.130
Dissonance	0.547 ± 0.077
Tonal stability	0.416 ± 0.065
Minorness	0.293 ± 0.083

5.1.3 Mid2E – From Mid-Level Features to Emotion

We next regress emotion ratings from the predicted mid-level features. These experiments were repeated six times to reduce variance.

Two modelling strategies are compared:

a) **Linear regression** — with and without second-order interaction terms;

b) **Shallow neural network** (architecture in Table 13; early stopping 15 epochs).

Table 12: Linear regression results with various feature sets.

Feature set	r	**MSE**
7 features, interactions	0.738 ± 0.022	1.629 ± 0.132
7 features, no interactions	0.733 ± 0.015	1.638 ± 0.084
19 features, interactions	0.525 ± 0.027	4.916 ± 0.905
19 features, no interactions	0.747 ± 0.012	1.566 ± 0.070
34 features, interactions	0.433 ± 0.015	5.545 ± 0.218
34 features, no interactions	0.734 ± 0.012	1.634 ± 0.062

Linear models Table 12 shows the results of the linear models in the prediction of emotions from various sets of mid-level faetures. For clarity, the "19 features" configuration simply appends the 12 mean chroma bins to the original 7 descriptors, while the 7-feature baseline uses the perceptual descriptors alone.
The "34 features" variant is made up of:

- the **7 mid-level descriptors** in Table 4;
- **24 binary chord flags**: presence of the 12 major and 12 minor triads predicted by `Essentia`;
- **key_root** — the tonic pitch class estimated by `Essentia`;
- **tonic_top3** — a Boolean that checks whether key_root matches one of the three most frequent chords in the excerpt;
- **key_mode** — the mode (major/minor) returned by `Essentia`, conceptually akin to the "minorness" perceptual descriptor.

Table 13: Architecture of the shallow Mid2E network.

Layer	**Description**
Input	Mid-level feature vector
Dense (32)	Fully-connected layer with 32 units
Dropout (0.3)	Applied to activations
Dense (8)	Output layer (8 emotion dimensions)

Table 14: Neural-network Mid2E results averaged over six seeds.

Feature set	r	**MSE**
7 mid-level	0.652 ± 0.093	2.066 ± 0.473
19 features	0.659 ± 0.084	2.027 ± 0.411
34 features	0.679 ± 0.073	1.959 ± 0.359

Neural network model Table 14 shows the results of the shallow neural network model in the prediction of emotions from the same sets of mid-level faetures described in the previous paragraph.

Despite dropout and early stopping, the neural variant underperforms the simpler linear regressors and shows larger run-to-run variance.

Mid2E: model choice over "more features" With many correlated inputs (mid-level descriptors, average chroma, chord flags), the linear model regularises the problem better than the shallow NN. The 7-feature baseline is already competitive; adding a compact harmonic summary (mean chroma $\rightarrow$ 19 features) gives a small, consistent bump. Turning on interactions on the larger 34-feature set actually hurts (Table 12), which looks like over-parameterisation relative to the data. The shallow NN also underperforms and shows higher variance even with dropout and early stopping (Table 14).

The improvement from 7 to 19 features in Table 12 is there, but not huge. Looking at the means and standard deviations in the tables, a compact representation of the harmonic content appears to provide some benefit. However, the improvement is small and would require further validation to confirm its significance.

Feature-importance analysis Below we present a unified analysis of feature importance across four configurations: 7 features with interactions, 7 features without interactions, 19 features, and 34 features, to offer a complete view of how information is distributed in both a linear Ridge model and a shallow neural network. The objective here is not only comparative performance, but *explainability*: tracing which descriptors matter, and in what direction, for a task that is simple to state (e.g., deciding whether a song is *happy* or *sad* from its spectrogram) yet acoustically

complex. To make cross–model and cross–setting comparisons meaningful, linear coefficients are reported in standardized space while neural importances are summarized with *global* SHAP (mean absolute attributions across emotions). Taken together, the heatmaps, ranked coefficients, and SHAP summaries provide complementary perspectives: where the linear model concentrates weight, where explicit interactions shift salience, and where the neural model distributes influence more diffusely. We will return to these figures in the following pages, highlighting convergences and divergences across the four setups and discussing what they suggest about linear structure versus (explicit or implicit) interactions.

7-Feature configuration: feature-importance visualizations In this section we focus on the 7 features setup. We show two linear models (one *without* interactions and one *with* pairwise interactions) and, separately, the shallow neural network. We include both linear variants because they both gave good predictive results in cross-validation and they let us look at the problem from two angles: the model without interactions shows the main effects of each descriptor; the model with interactions shows how pairs of descriptors work together. For all linear plots the coefficients are in standardized space (a standard standardization was applied) and we keep the same, symmetric colour scale across emotions. Next to each heatmap we also report a bar plot that ranks features by the mean absolute weight across the eight emotions. For the neural network we use global SHAP (the mean of SHAP values attributions aggregated over emotions) to summarize importance in a single view.

Main effects (no interactions). The baseline linear model (Figure 10, Figure 11) is quite readable. `melodiousness` shows positive weights for *valence*, *happy*, *sad* and *tender*, and negative weights for *tension*, *anger*, and *fear*. `dissonance` behaves in the opposite way (down for *valence*/*sad*, up for *tension*/*anger*/*fear*), which fits the idea that dissonance adds tension. `articulation` is strongly linked to *energy* (and somewhat to *tension*/*anger*, and relates negatively to *tender*). `minorness` separates positive and negative affect in the expected major/minor direction. The ranking confirms a small core of important variables: melodiousness, minorness, dissonance and articulation, while the two rhythm descriptors play a smaller role.

Adding interactions. When we add explicit pairwise terms (Figure 12, Figure 13), the overall picture does not change, but some importance moves from single features to specific pairs. Several combinations with `tonal_stability` become visible in the top part of the ranking (for example `articulation + tonal_stability`, `rhythm_complexity + tonal_stability`, `tonal_stability + minorness`). This suggests that the effect of tonal_stability depends on how notes are articulated and organised in time. In short, interactions add a degree of complexity, but the same few descriptors remain the main drivers.

Neural model (SHAP). The SHAP summary for the shallow network (Figure 14) is broadly consistent with the linear view. The largest contributions still come from

`melodiousness`, `articulation`, `dissonance`, and `minorness`, with smaller contributions from `tonal_stability` and rhythm. Compared to the linear model, `articulation` ranks higher in the neural network. We think the network is picking up cues about how notes start and change in loudness over time (onsets and envelopes), which are non-linear and hard for a linear model to capture. Also, the SHAP distribution is more spread out, meaning the network distributes importance across more features instead of putting very large weights on a few of them.

Takeaways. (i) A small, musically meaningful set—`melodiousness`, `minorness`, `dissonance`, `articulation`—explains most of the behaviour in both models. (ii) Interactions mainly highlight conditional effects with `tonal_stability` but do not overturn the ranking. (iii) The neural model agrees with the same cues, while allocating importance more smoothly. For these reasons we keep both versions (with and without interaction) variants in the 7-feature analysis.

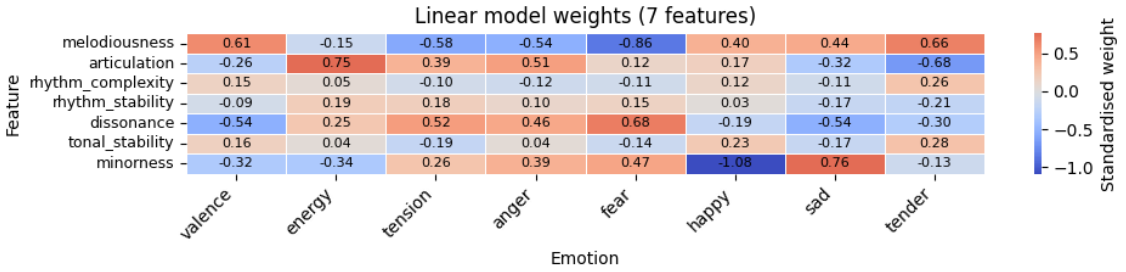


Figure 10: Normalised weights of the 7-feature linear regression without interaction (only first-order terms).

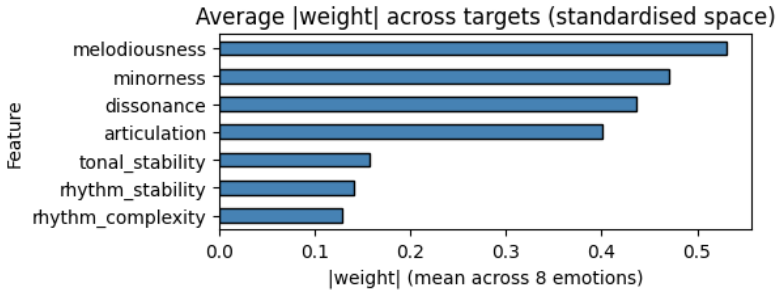


Figure 11: Absolute weight ranking for the same 7-feature linear model (no interactions).

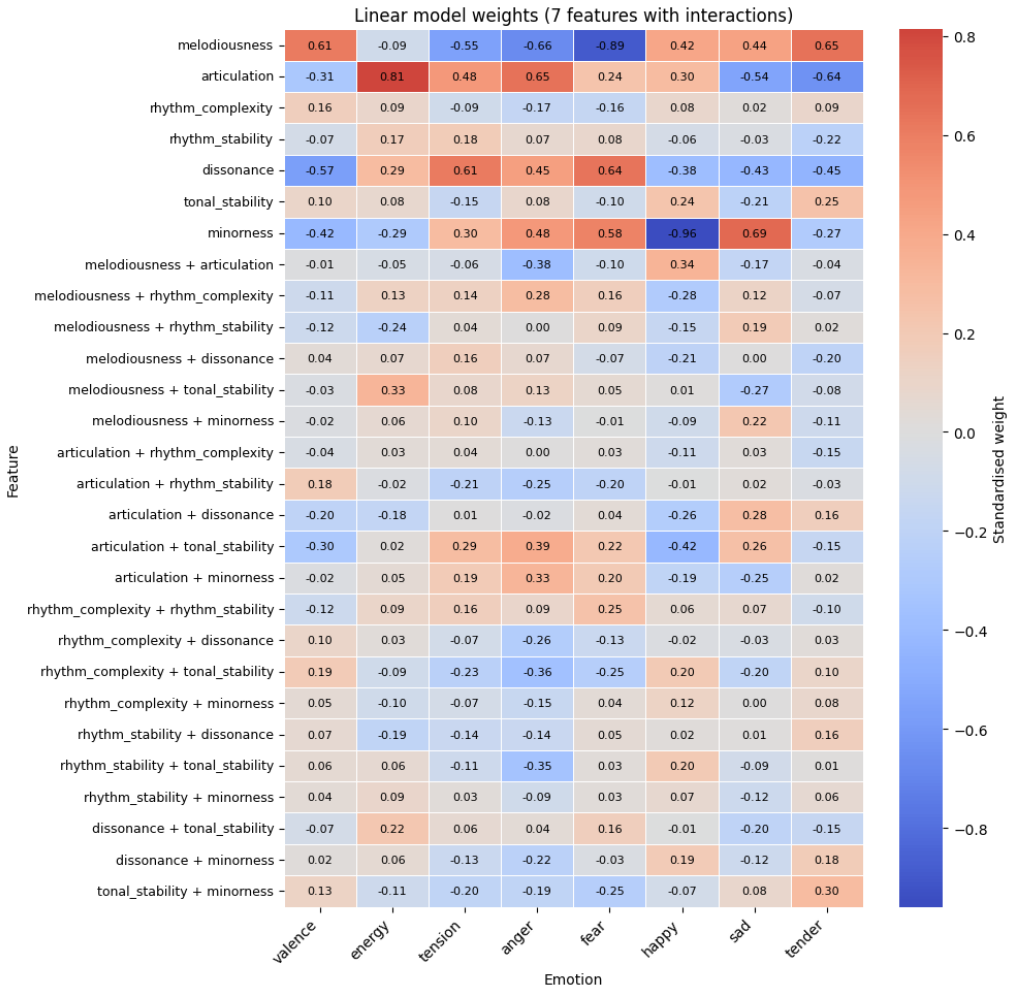


Figure 12: Normalised weights of the 7-feature linear regression with interaction (up to second order).

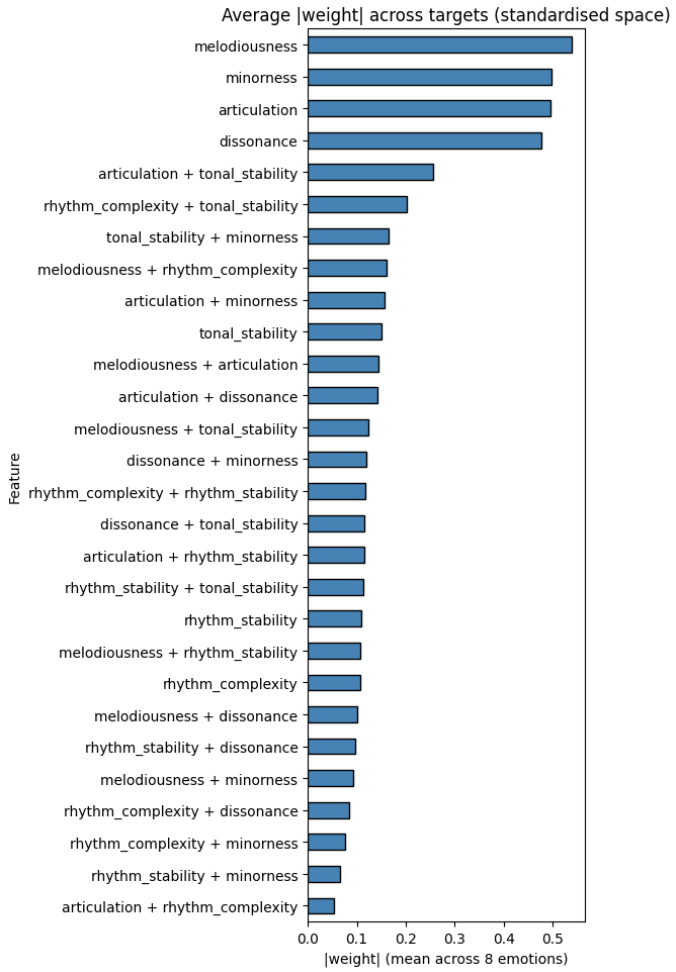


Figure 13: Absolute weight ranking for the same 7-feature linear model (with interactions).

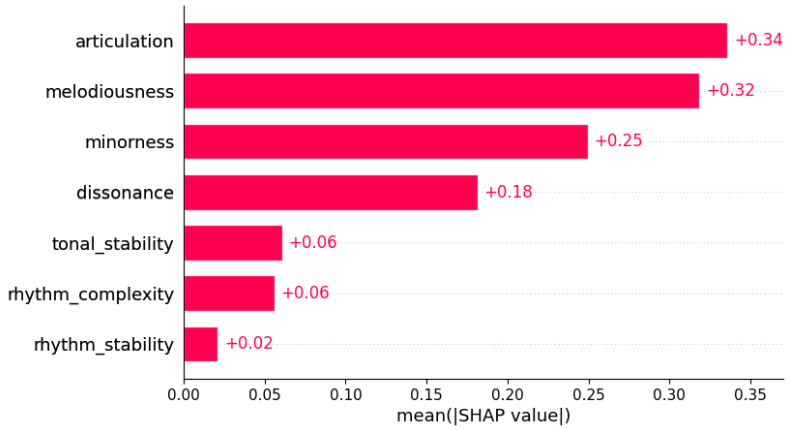


Figure 14: Mean SHAP values for the shallow neural network with 7 features.

19-Feature configuration: feature-importance visualizations Here we analyse the 19 features setup. We only report the linear model without interactions and the shallow neural network. As discussed in the introduction of this section, we do not include interaction-augmented versions for larger sets because early trials were not convincing and the number of terms grows quickly, which goes against our explainability goal. As before, linear coefficients are shown in standardized space with a common, symmetric colour scale, and we add a bar plot that ranks features by the mean absolute weight across the eight emotions. For the neural model we use again global SHAP.

Linear model (no interactions). The heatmap in Figure 15 shows patterns that are consistent with the 7-feature case, but with some extra detail coming from pitch-class features. The strongest and clearest effects are the ones following. `minorness` has a marked polarity: negative for *valence* and especially *happy*, and positive for *sad*, *fear*, and *anger*. `dissonance` is negative for *valence*, *sad* and *tender* and positive for *tension*, *anger*, and *fear*. `melodiousness` goes in the opposite direction: positive for *valence*/*happy*/*tender*. `articulation` is strongly linked with *energy* and shows smaller positive weights for *tension*/*anger*, and a negative relation with *tender*.

Pitch class features (C, C#/Db, . . . , B) are weaker on average, but a few of them (e.g., F, G#/Ab, A) show consistent mid-level contributions across several emotions. A possible explanation is that, given the small dimension of the dataset, some specific pitch classes were learned overfitting on various pieces of the same song. This overfitting could be one of the reasons why using more complex sets of features does not provide a large margin of improvement. Exploring larger datasets and using a relative encoding could make the difference and in that case more complex sets of features could have better performance. The ranking in Figure 16 confirms this picture: the top group is `minorness`, `dissonance`, `melodiousness`, and `articulation`; then we find some pitch classes and `tonal_stability`; rhythm descriptors remain in the lower half.

Neural model (SHAP). The SHAP summary in Figure 17 broadly agrees with the linear results but changes the order of the top features: `melodiousness` comes first, followed by `articulation`, `minorness`, and `dissonance`. This suggests that the network gives more value to cues related to how notes start and how their loudness

evolves over time (onsets and amplitude envelopes), which are non-linear and harder for a linear model to capture. SHAP also brings a few pitch classes (for example F, G#/Ab, A) slightly higher than in the linear ranking, while `rhythm_complexity` and `rhythm_stability` stay relatively small in both models. Overall, the neural importance is more spread out: the network distributes weight across more descriptors instead of relying on a few very large coefficients.

Takeaways. (i) The core cues seen with 7 features—`minorness`, `dissonance`, `melodiousness`, and `articulation`—remain dominant with 19 features. (ii) Adding pitch-class information improves the granularity of the linear model without changing the global picture; some pitch classes enter the mid-range of the ranking but do not replace the main four cues. (iii) The neural model keeps the same set of important descriptors but pushes `articulation` to the top and spreads importance more smoothly. This supports the idea that both models agree on *what* matters, while they differ a bit on *how* these cues are used.

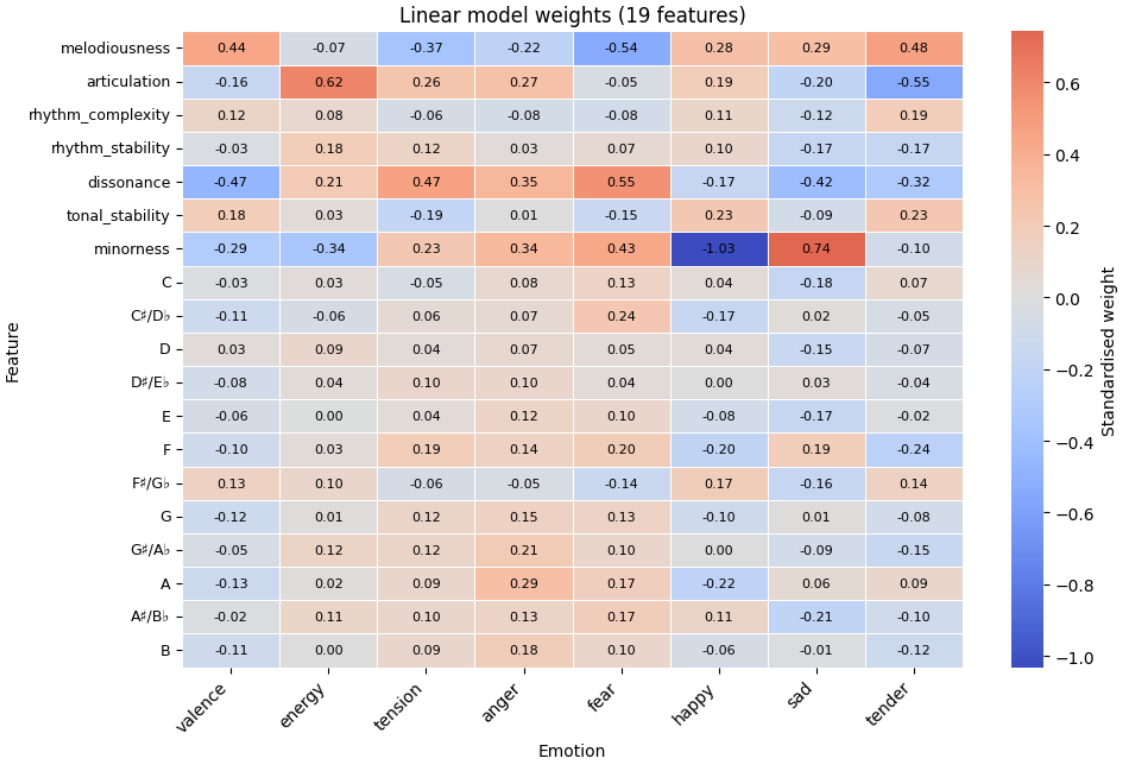


Figure 15: Normalised weights of the 19-feature linear regression without interactions.

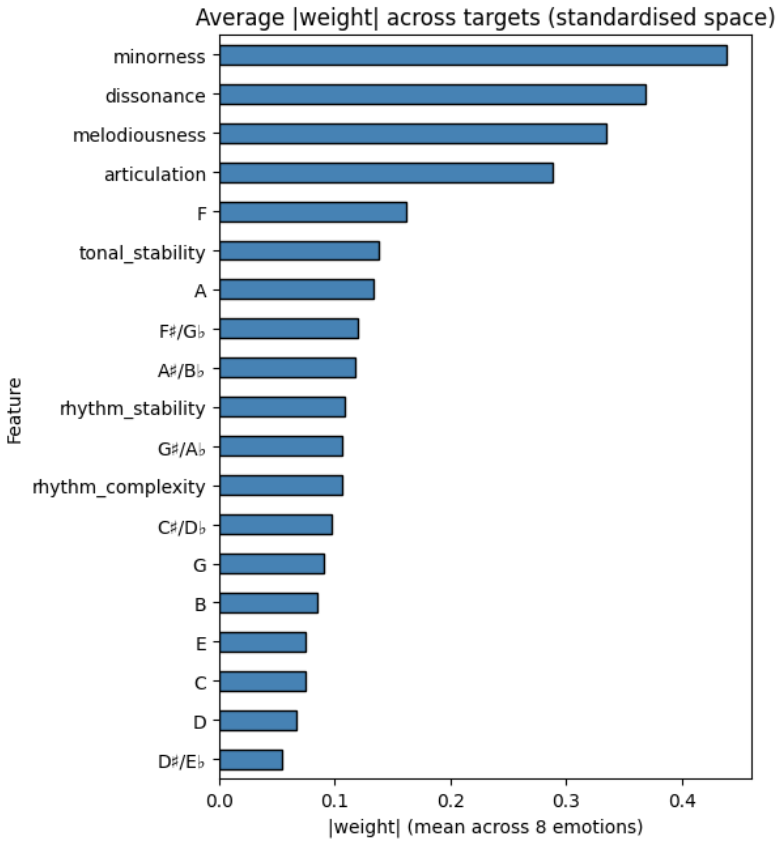


Figure 16: Absolute weight ranking for the same 19-feature linear model (no interactions).

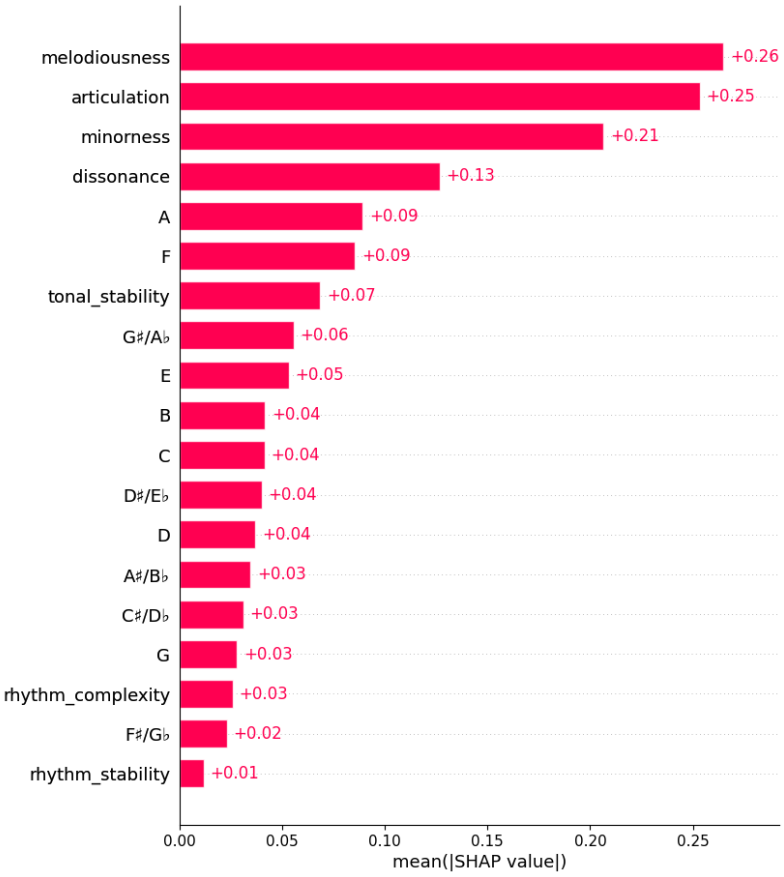


Figure 17: Mean SHAP values for the shallow neural network with 19 features.

34-Feature configuration: feature-importance visualizations Here we analyse the 34 features setup. As for the 19-feature case, we only report the linear model without interactions and the shallow neural network. Interaction-augmented versions were not included because early tests were not very convincing and the number of terms would become very large, which is not ideal for explainability. We keep the same plotting choices as before: linear coefficients in standardized space with a common, symmetric colour scale, a bar plot that ranks features by the mean absolute weight across the eight emotions, and global SHAP for the neural model.

Linear model (no interactions). The heatmap in Figure 18 extends what we saw with 19 features. The four main cues remain very clear: `melodiousness` is positive for *valence*, *happy*, *sad*, and *tender*, and negative for *tension*, *anger*, and *fear*; `dissonance` shows the complementary pattern (down for *valence*/*happy*, up for *tension*/*anger*/*fear*); `articulation` aligns strongly with *energy* and has smaller positive links to *tension*/*anger* and negative links to *tender*; `minorness` keeps the expected major/minor polarity (negative for *valence*/*happy*, positive for *sad*/*fear*/*anger*). With 34 features we also include key-related descriptors: `key_mode` and `key_root` behave in a way that is consistent with `minorness`, while individual pitch classes and minor/major keys show small but coherent weights in several columns. The ranking in Figure 19 reflects this picture: the top group is `melodiousness`, `dissonance`, `articulation`, and `minorness`; then we see `tonal_stability`, `key_mode`, and a subset of pitch-class/key features; rhythm descriptors remain lower (`rhythm_stability` above `rhythm_complexity`).

Neural model (SHAP). The SHAP summary in Figure 20 mostly agrees with the linear model but changes the order of the top features. `articulation` comes first, followed by `melodiousness`, then `minorness` and `dissonance`. This suggests that the network gives extra value to cues about how notes start and how loudness evolves over time (onsets and amplitude envelopes), which are non-linear and harder for a linear model to capture. We also see key-related features (`key_mode`, `key_root`) appearing in the top ten. Several pitch classes enter with small contributions (e.g., D, E, B, Eb), but each one is modest on its own. The last bar ("sum of 15 other features") makes clear that, with 34 inputs, the neural importance becomes more spread out across many small terms.

Takeaways. (i) The same core cues—`melodiousness`, `dissonance`, `articulation`, `minorness`—remain dominant even after adding many new descriptors. (ii) Key-related features add useful granularity and are consistent with the global `minorness` signal, while individual pitch classes/keys contribute in a mild but coherent way. (iii) The neural model keeps the same set of important descriptors but pushes `articulation` to the top and distributes importance more smoothly over the long tail. Overall, both models agree on *what* matters, and they differ slightly on *how* these cues are used.

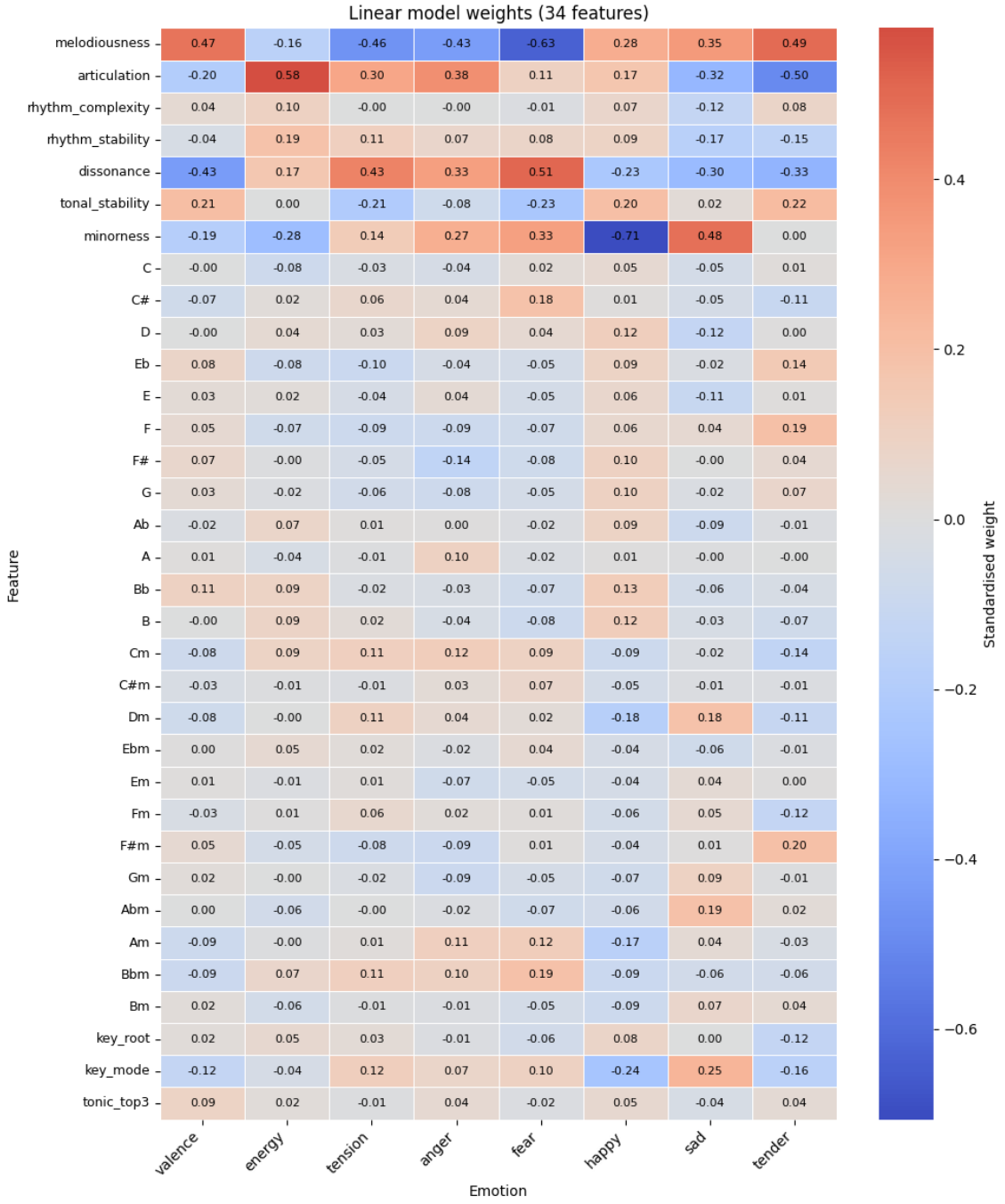


Figure 18: Normalised weights of the 34-feature linear regression without interactions.

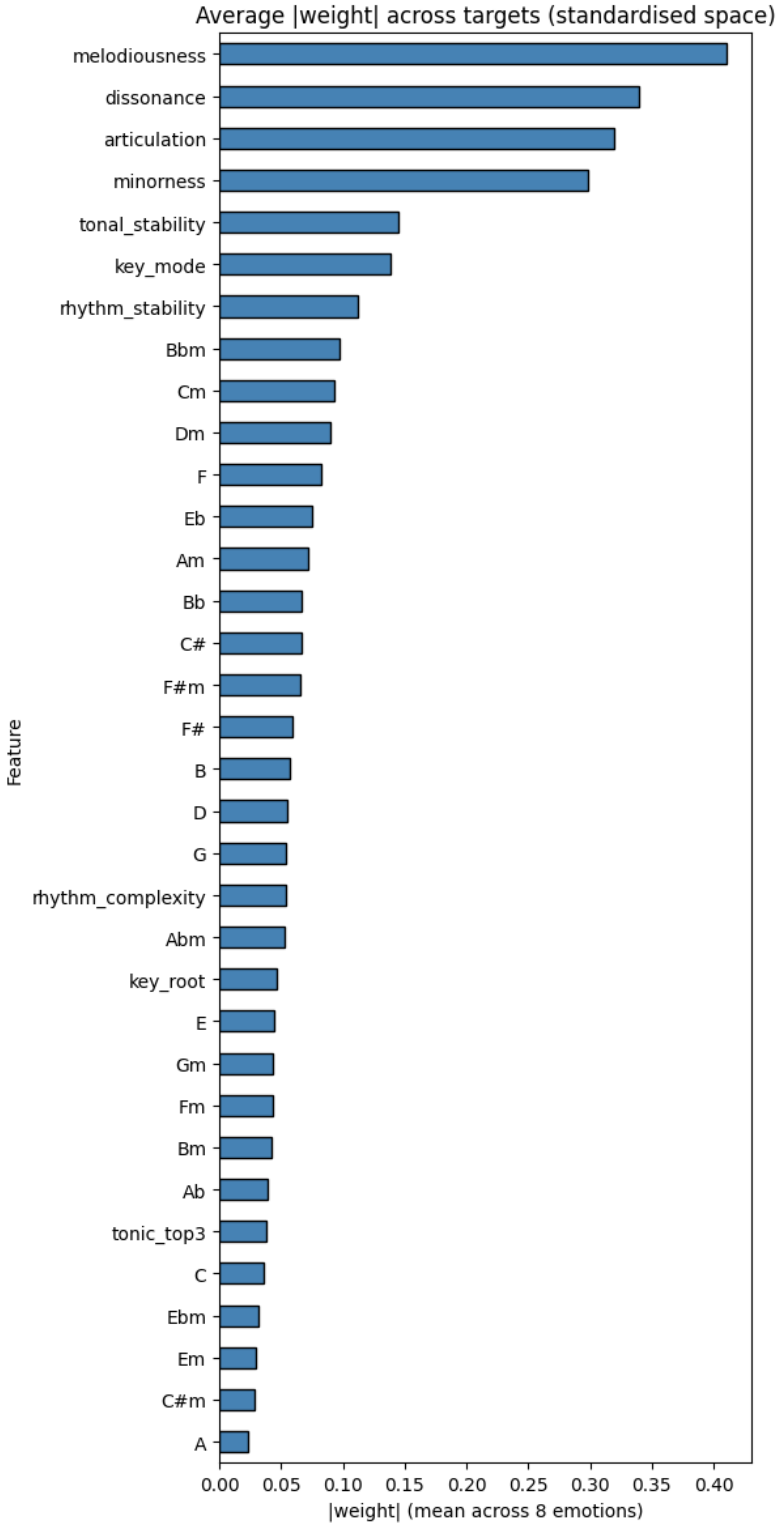


Figure 19: Absolute weight ranking for the same 34-feature linear model (no interactions).

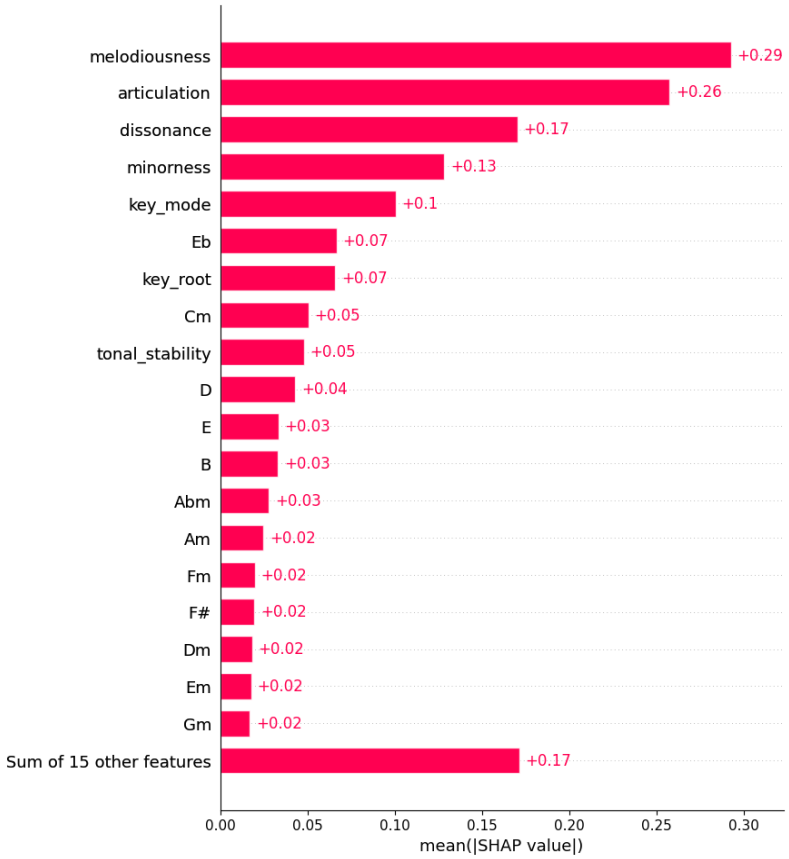


Figure 20: Mean SHAP values for the shallow neural network with 34 features.

5.2 Experiments on sonification

This section describes how we turn the LRP attribution maps from our fine-tuned ViT classifier into simple sonifications. Here we switch focus to the Musical Emotions Classification dataset (Section 4.2.3), where the ViT backbone (Section 2.4.2) is fine-tuned to distinguish "Happy" vs. "Sad" audio excerpts. By computing attributions on spectrogram inputs, we reveal which time–frequency regions the model attends to—and then render those regions as audible cues. In the GitHub repository of the project you can listen to the sonification examples presented in this section.

5.2.1 Setup

The ViT is fine-tuned for 60 epochs (learning rate 1e-4, batch size 8) with early stopping (patience=10). Across different seeds, convergence typically occurs within 5–10 epochs. To mitigate the class imbalance (135 "Sad" vs. 102 "Happy" examples), we employ a weighted cross-entropy loss: each class's loss contribution is scaled

inversely proportional to its number of samples, giving the minority "Happy" class a larger weight during training. The final model achieves a test accuracy of `0.776` on held-out test data.

5.2.2 Classification Results and Confusion Matrix

After fine-tuning, the ViT classifier reaches an overall test accuracy of 0.776. To better understand its behaviour on each emotion class, we examine the confusion matrix in Figure 21 and report per-class precision and recall.

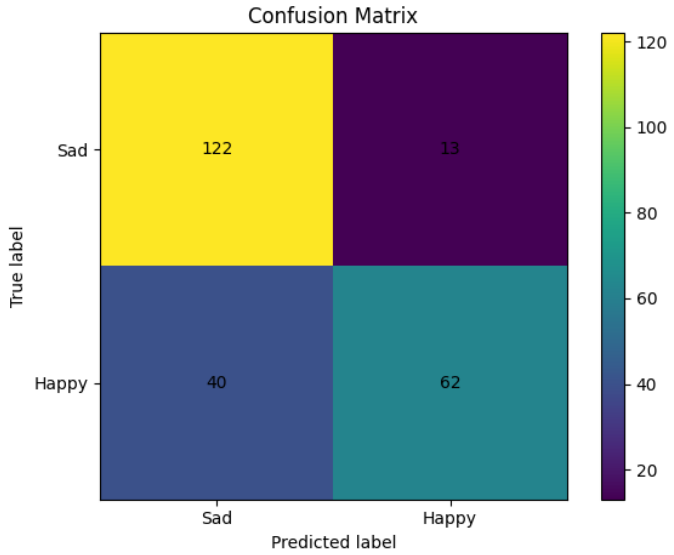


Figure 21: Confusion matrix for the "Happy" vs. "Sad" classification on the test set. Rows correspond to true labels, columns to predicted labels.

Figure 21 details per-class behaviour. Of 135 true "Sad" excerpts, 122 are correctly identified (recall $= 122/135 \approx 0.90$); of 102 true "Happy" excerpts, 62 are correctly classified (recall $= 62/102 \approx 0.61$). Precision is $122/(122 + 40) \approx 0.75$ for Sad and $62/(62 + 13) \approx 0.83$ for Happy. Macro-averaged precision and recall are 0.79 and 0.76, and the weighted F_1 is 0.77. In practice, the model is sensitive to "Sad" (high recall) but tends to be conservative in predicting "Happy," leading to higher precision but lower sensitivity for that class. This matches the class imbalance (135 vs. 102) and the idea that some acoustic cues are shared between the two labels.

5.2.3 Attribution map extraction and visualization

To understand those errors, we will look at four examples, one for each cell fo the confusion matrix: Happy→Happy (Fig. 22), Happy→Sad (Fig. 23), Sad→Sad (Fig. 24), and Sad→Happy (Fig. 25).

We first compute log-magnitude spectrograms at a 22.05kHz sampling rate using an FFT size of 3198 samples ($\approx 145ms$ window) and hop length of 989 samples ($\approx 45ms$), yielding 1600×224 grayscale images. To satisfy the ViT's 224×224 input requirement (Section 2.4.2), we downsample along the frequency axis to 224×224 before inference. LRP attributions are then computed on these 224×224 inputs and then upsampled back to 1600×224, producing full-resolution relevance masks $R(\tau, f)$.

We retain the complex STFT throughout so that during sonification we can invert the spectrum with Librosa's `istft` using the original phase.

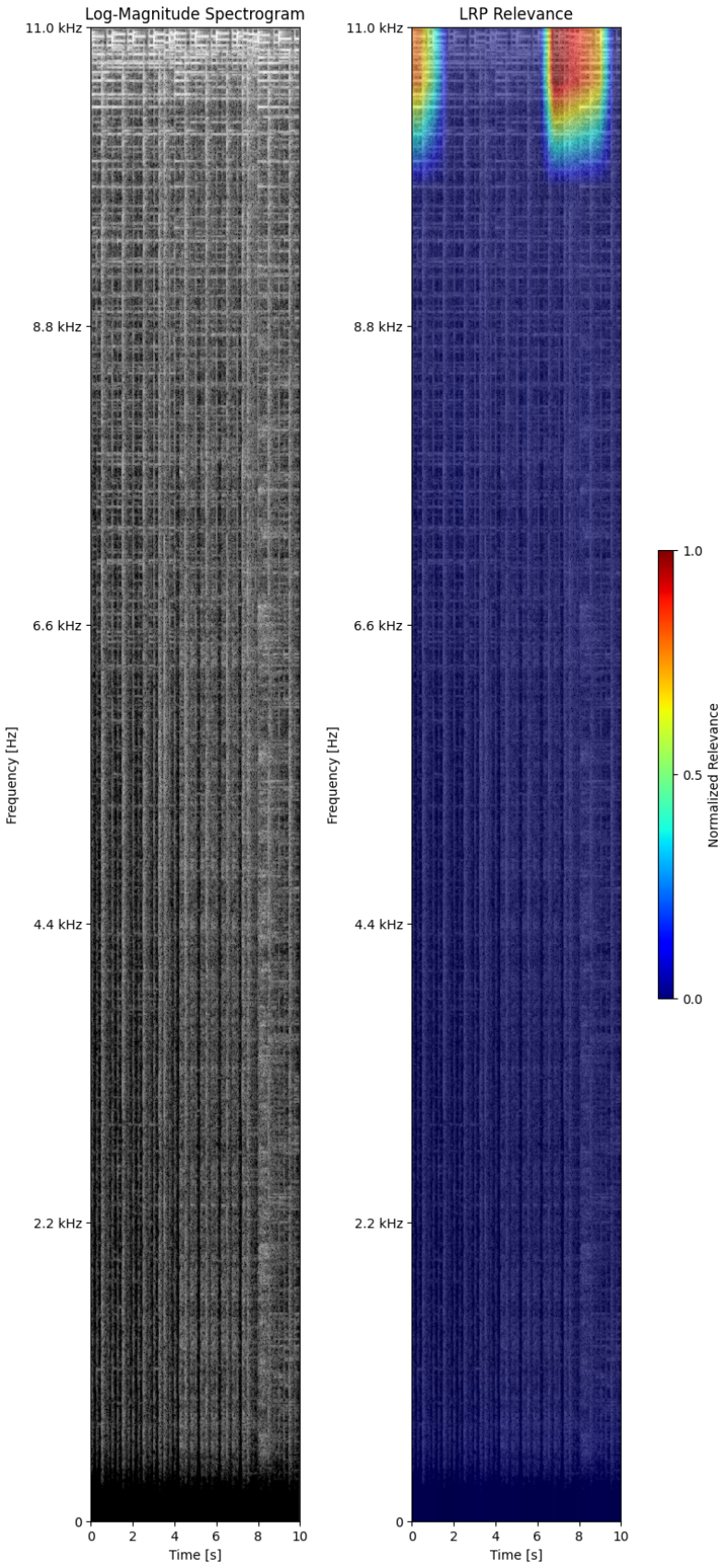


Figure 22: Attribution map for a correctly classified "Happy" excerpt (99.61% confidence).
Left: spectrogram.
Right: upsampled LRP map; warmer colors = higher contribution.

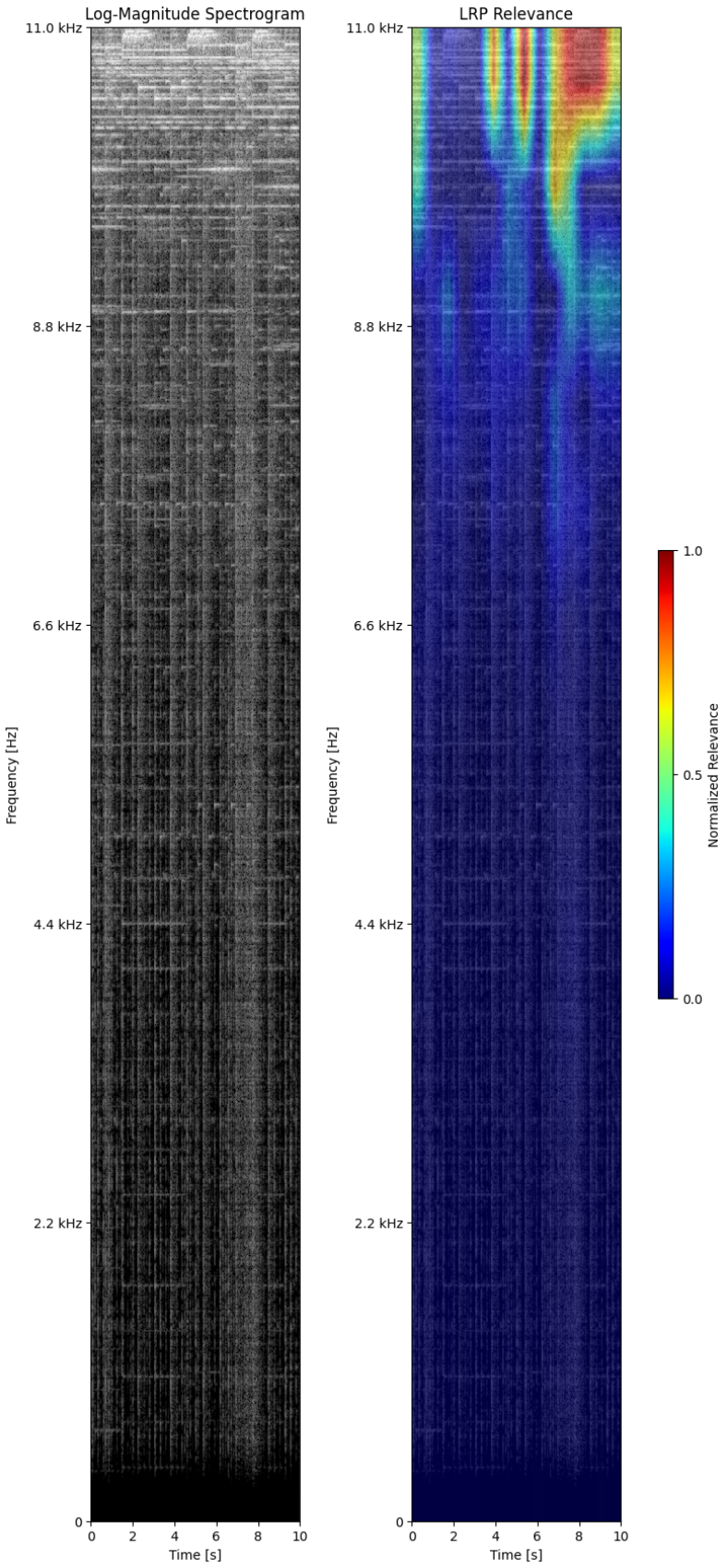


Figure 23: Attribution map for a wrongly classified "Happy" excerpt (predicted sad with 85.23% confidence).
Left: spectrogram.
Right: upsampled LRP map; warmer colors = higher contribution.

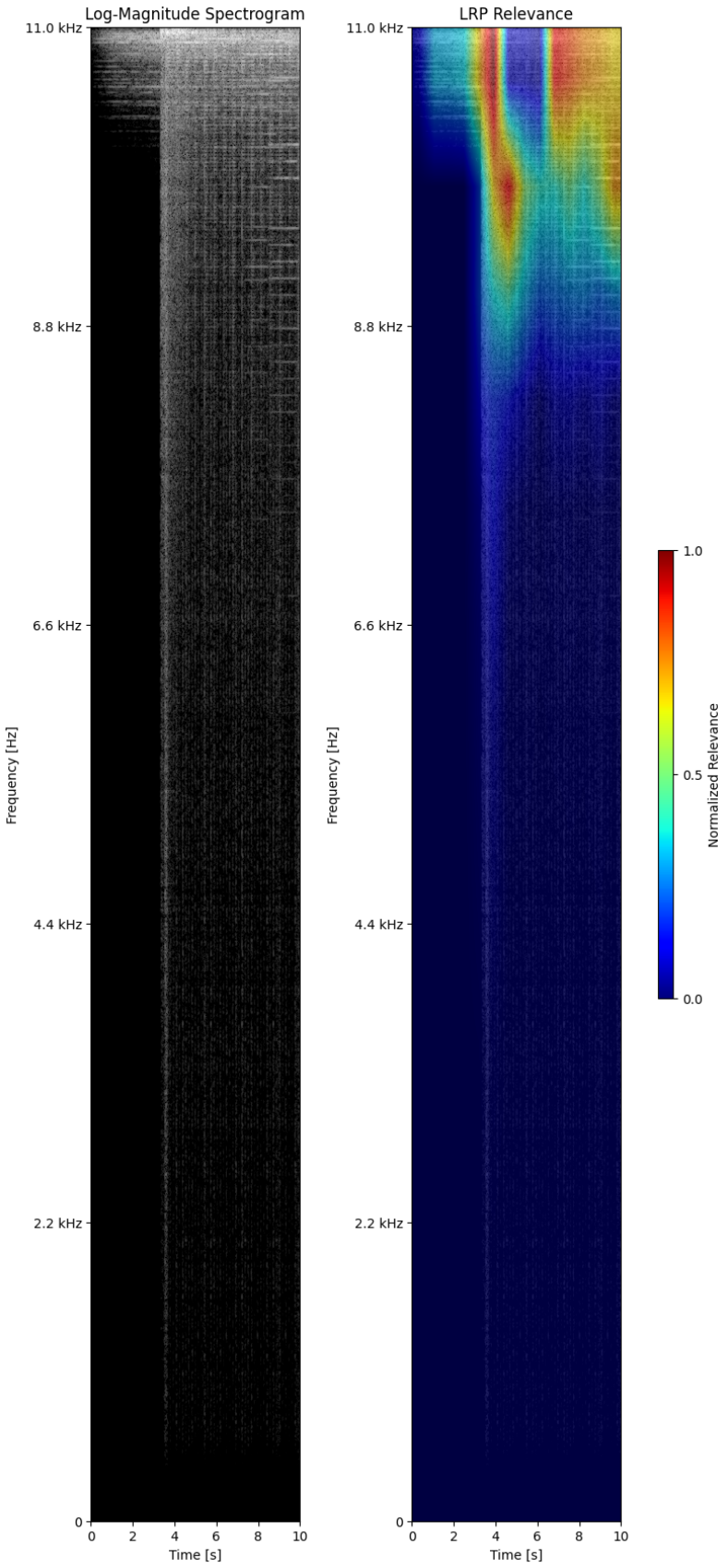


Figure 24: Attribution map for a correctly classified "Sad" excerpt (93.17% confidence).
Left: spectrogram.
Right: upsampled LRP map; warmer colors = higher contribution.

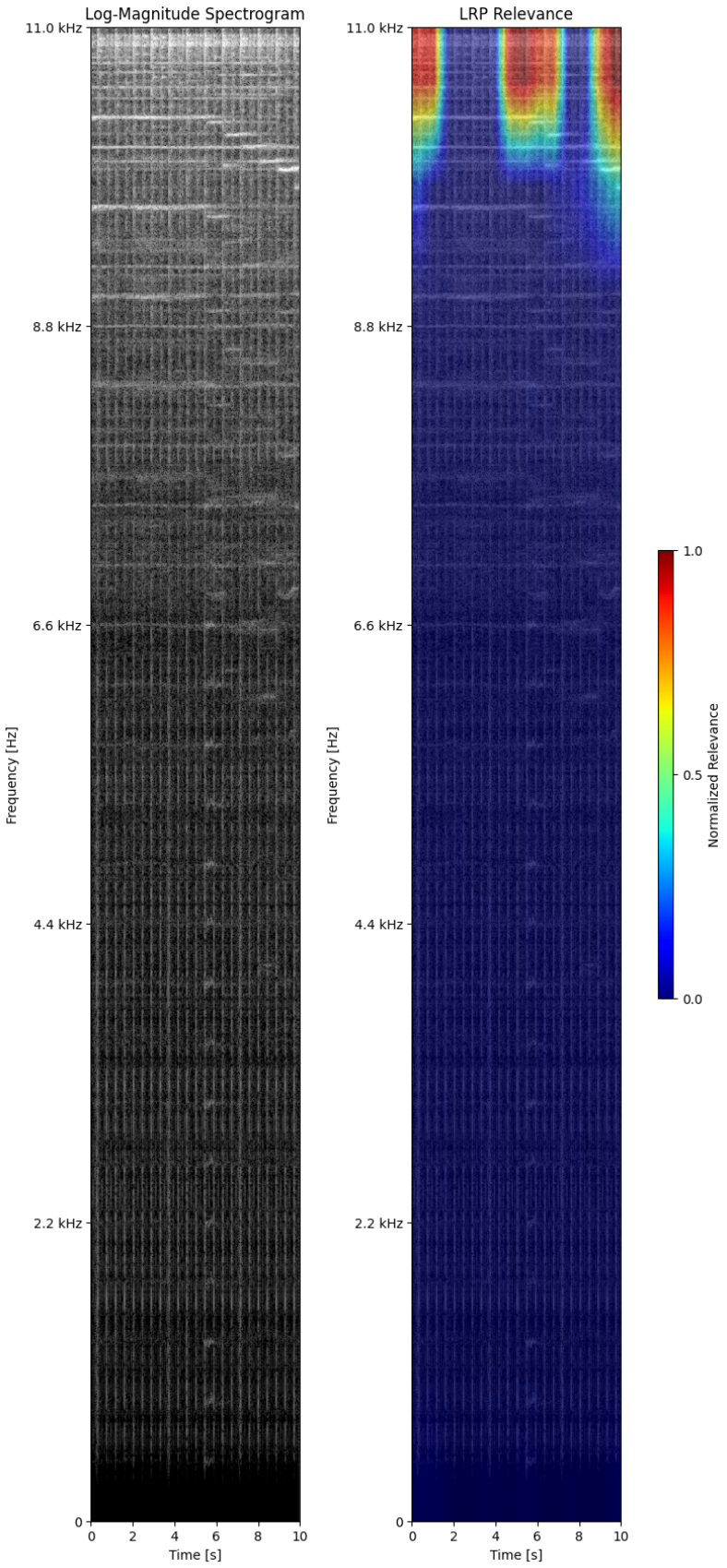


Figure 25: Attribution map for a wrongly classified "Sad" excerpt (predicted happy with 89.88% confidence).
Left: spectrogram.
Right: upsampled LRP map; warmer colors = higher contribution.

5.2.4 Sonification mapping strategy

To generate the sonification, we weight each complex STFT coefficient $X(\tau, f)$ by the squared attribution mask, following Eq. (15) with a quadratic gain:

$$\widehat{X}(\tau, f) \;=\; R(\tau, f)^2\, X(\tau, f). \tag{24}$$

This boosts high-relevance bins so that, when listening to the sonification, the relevance is even more accentuated. We then apply Librosa's `istft` to $\widehat{X}$, reconstructing a waveform that audibly emphasizes the model's focus.

5.2.5 What the examples suggest

Our reading is that the model has associated a regular, faster rhythmic surface with Happy. This helps when the target is indeed Happy and rhythm is present (Fig. 22), but it also explains the two error types:

- **Happy→Sad.** When a Happy clip is more relaxed or sparser, the highlighted regions are weaker and the model seems to default to Sad (Fig. 23).
- **Sad→Happy.** When a Sad clip has pronounced, steady percussion (e.g., regular drum hits), the map focuses on those onsets and the model flips to Happy (Fig. 25).

This is consistent with what we hear in many tracks: rhythm alone can bias the impression. With a small dataset, and with several clips cut from the same songs, the model probably learned the most frequent, easy cues first (steady pulse for Happy; lower, sustained energy for Sad), and did not see enough counter-examples to refine that rule.

6 Conclusion and Future Work

The main idea behind this thesis was simple: put explainability first. Rather than only chasing higher scores, this work tried to build and analyse models that we can read, question, and learn from. This matters now more than ever. In research and in everyday tools, people need to know why a model behaves a certain way, not just whether it is accurate on average. As AI systems get closer to human performance, the difference will be less about "who gets the best number" and more about "who can be understood, trusted, and improved." For music in particular, explanations that connect to musical structure, or that you can literally listen to, are far more useful than a raw metric.

6.1 Conclusions

Looking back at the experiments, these are the observations that appear:

- **Explanations allow to see what was each feature's contribution.** With linear coefficients and SHAP it was possible to check which inputs actually mattered (and when extra features just added noise). This made over-parameterisation visible and helped preferring simpler, more stable heads when features were correlated.
- **Sonifications made the reasoning clearer.** The LRP-based sonification turned a hard-to-read image into something that it was possible to listen to. Hearing the boosted onsets/patterns made it much clearer what the classifier was putting its focus on and why certain mistakes happened.
- **Explanations changed the research decisions.** They weren't just "nice plots": they guided modelling choices (e.g., avoid large interaction sets, keep harmonic summaries compact), flagged dataset biases (steady rhythm → "Happy"), and suggested concrete fixes (threshold calibration, using weights for Happy examples). In short, they turned the model into a tool for hypothesis testing rather than just a score generator.

In short, explainability was not an add-on at the end; it shaped the modelling choices

and the way the results were interpreted.

6.2 Future Work

The most important next step is straightforward: **better data**. A larger and more diverse corpus (balanced classes, fewer near-duplicates from the same tracks, and ideally labels beyond a binary split) would let the models rely on richer features and would make the explanations more interesting. It would also make more possible to learn from extracted features, as with more training data it would be possible to recognize more patterns and reduce the variance typical of extracted features.

Beyond data, a few other directions:

- **Key–relative encodings of musical structure.** Musical information is mostly relative (to key, function, interval). Instead of raw chord one-hots, it would be better trying key–relative or function-of-harmony encodings, interval patterns, or low-rank harmonic embeddings that are transposition-invariant.
- **More trustworthy models and checks.** Explore models and techniques aimed at reliability: compare attribution methods and run sanity checks; apply more XAI methods to see how this changes our perception of the explanation.
- **Apply to real-world catalogues and use-cases.** Test the approach on mainstream music (popular genres, chart tracks) and genre-specific subsets, where production cues and conventions differ. Tie explanations to practical applications (e.g., recommendation "rationales", playlisting, creative tools) and run small user studies to see if people find the explanations helpful. This would make the work more directly useful and highlight domain shifts that do not appear in small academic datasets.

Overall, the thesis shows that, even when working with musical emotions, starting from explainability is a workable strategy: it helps build models you can reason about, it surfaces limitations early, and it points clearly to what to fix next. With better datasets and a few careful design choices, the same approach should scale to richer emotions and more realistic musical material.

7 Glossary

MER (Music Emotion Recognition)

Task of predicting the emotional content of a musical excerpt based on its audio signal, either categorical (Happy/Sad) or dimensional (valence/arousal).

Mid-level Features

Perceptual descriptors (e.g., melodiousness, rhythmic stability, dissonance) that mediate between raw audio and emotion labels.

Spectrogram

Time–frequency representation of audio, showing how spectral energy evolves over time.

Chromagram (Chroma Features)

Representation condensing spectral energy into 12 pitch classes (C–B), regardless of octave, focusing on harmonic/tonal structure.

Chord / Chord Progression

Simultaneous combination of notes and their sequential arrangement, often linked to harmonic and emotional perception.

CNN (Convolutional Neural Network)

Deep learning architecture using convolutional filters to detect patterns in data like spectrograms.

ViT (Vision Transformer)

Deep model processing spectrograms as sequences of patches through self-attention, enabling long-range dependencies.

SHAP (SHapley Additive exPlanations)

Game-theoretic explainability method assigning each feature a contribution value to the prediction.

LRP (Layer-wise Relevance Propagation)

Attribution technique redistributing the model's output backwards through its layers to highlight relevant input parts.

Sonification

Transformation of model attribution maps into sound, so that explanations can be perceived aurally.

Accuracy

Classification metric: ratio between correct predictions and total predictions.

Precision

Proportion of predicted positives that are truly positive.

Recall (Sensitivity)

Proportion of true positives correctly identified by the model.

F1 Score

Harmonic mean of precision and recall, balancing false positives and false negatives.

MSE (Mean Squared Error)

Regression metric: average squared difference between predicted and true values.

Pearson's r

Statistical measure of linear correlation between predicted and true values.

Cross-Entropy Loss

Classification loss function penalizing the divergence between predicted probability distribution and true class.

References

Aljanaki, A., & Soleymani, M. (2018a). A data-driven approach to mid-level perceptual musical feature modeling. *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 615–621. https://github.com/MTG/midlevel-emotion

Aljanaki, A., & Soleymani, M. (2018b). Emotion recognition in music using audio-visual features and multiple classifiers. *Proc. of the MediaEval Workshop*.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, *10*(7). https://doi.org/10.1371/journal.pone.0130140

Chefer, H., Gur, S., & Wolf, L. (2021). Transformer interpretability beyond attention visualization. *CVPR*, 782–791.

Choi, K., Fazekas, G., Cho, K., & Sandler, M. (2017). Convolutional recurrent neural networks for music classification. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2392–2396.

Chowdhury, S., Vall, A., Haunschmid, V., & Widmer, G. (2019). Towards explainable music emotion recognition: The route via mid-level features. *Proc. of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 257–264.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*.

Dubus, G., & Bresin, R. (2013). A systematic review of mapping strategies for the sonification of physical quantities. *PLOS ONE*, *8*(12), e82491. https://doi.org/10.1371/journal.pone.0082491

Eerola, T., & Vuoskoski, J. K. (2011). A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, *39*(1), 18–49. https://doi.org/10.1177/0305735610362821

Er, M., & Aydilek, İ. (2019). Music emotion recognition by using chroma spectrogram and deep visual features. *International Journal of Computational Intelligence Systems*, *12*. https://doi.org/10.2991/ijcis.d.191216.001

Friberg, A., Schoonderwaldt, E., Hedblad, A., Fabiani, M., & Elowsson, G. (2014). Using listener-based perceptual features as intermediate representations in music information retrieval. *The Journal of the Acoustical Society of America*, *136*(4), 1951–1963.

Gresham, G., Pike, B., & Keller, B. (2020). Auditory displays for machine learning: Sonification of explanations for deep convolutional neural networks. *Proceedings of the 26th International Conference on Auditory Display (ICAD)*. https://icad2020.icad.org/pdfs/ICAD2020_paper_19.pdf

Grond, F., & Berger, J. (2011). Parameter mapping sonification. In T. Hermann, A. Hunt, & J. G. Neuhoff (Eds.), *The sonification handbook* (pp. 363–397). Logos Verlag. https://sonification.de/handbook

Haque, K. N. (2023, February 1). *What is convolutional neural network — cnn (deep learning)* [Accessed: 2025-06-13]. https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, *30*.

Montavon, G., Samek, W., & Müller, K.-R. (2019). *Explainable ai: Interpreting, explaining and visualizing deep learning*. Springer.

Musical emotions classification [Available at: urlhttps://www.kaggle.com/datasets/kingofarmy/musical-emotions-classification]. (2020).

Papermaker AI. (2024). *Explainable AI*. Retrieved June 20, 2025, from https://papermaker.ai/explainable-ai/

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Zhang, K., Zhang, H., Li, S., Yang, C., & Sun, L. (2018). The pmemo dataset for music emotion recognition. *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 135–142. https://doi.org/10.1145/3206025.3206037

Zhang, Y., Wang, Y., & Yang, Y.-H. (2018). Pmemo: A dataset for music emotion recognition with playlist context. *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 135–142.

Zohar, Y., Perry, O., & Soudry, D. (2021). Audiolime: Reproducing explainability of audio classifiers. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.