



## **Transforming musical audio data into visual art**

A study of machine learning methods to simulate Wassily Kandinsky's process of creating visual art from sounds.

Jaana Moilanen

Master's Thesis  
Master of Engineering - Big Data Analytics  
April 7, 2022

<b>MASTER'S THESIS</b>	
Arcada University of Applied Sciences	
Degree Programme:	Master of Engineering - Big Data Analytics
Identification number:	8646
Author:	Jaana Moilanen
Title:	Transforming musical audio data into visual art A study of machine learning methods to simulate Wassily Kandinsky's process of creating visual art from sounds.
Supervisor (Arcada):	Leonardo Espinosa-Leal, Ph.D.
Commissioned by:	
<p><b>Abstract:</b></p> <p>Data is used in nearly all aspects of life, and is becoming more and more advanced. There are machine learning models and algorithms used in advancing human life, in modifying behaviour, and protecting data from harm. This research is focused on finding out how to utilise data in advancing something completely human; art.</p> <p>Data has been traditionally connected to parts of the professional world where we can discern facts. However, applying data to art is immensely more difficult because art can't necessarily be categorised into specific facts. Its very nature is highly subjective to the artist's vision and talent, but also to the perception of its audience.</p> <p>To understand how to use musical data to create visual art, we have researched ways to use music and art theory in combination with machine learning methods. This research studies audio source separation and feature extraction, and utilises available audio data visualisation libraries.</p> <p>To create a final visual composition, the results from these methods were combined with the colour and form theories of the abstract painter Wassily Kandinsky.</p>	
Keywords:	audio data visualisation, audio classification
Number of pages:	75
Language:	English
Date of acceptance:	07.04.2022

# CONTENTS

<b>Definitions</b>	<b>7</b>
<b>1 Introduction</b>	<b>12</b>
<b>2 Related Work</b>	<b>14</b>
2.1 Structural analysis of recorded music	14
2.2 Investigating audio data visualization for interactive sound recognition	15
2.3 A Survey on Visualizations for Musical Data	16
2.4 Audio Event Classification Using Deep Learning Methods	16
2.5 Visualization in Audio-Based Music Information Retrieval	16
2.6 Hybrid Spectrogram and Waveform Source Separation	17
<b>3 Research Methodology</b>	<b>18</b>
3.1 Audio classification	18
3.2 Audio visualization libraries	18
3.3 Visualization of musical audio data	19
<b>4 Experiments</b>	<b>21</b>
4.1 MusiCNN and VGG	21
4.2 Spleeter	22
4.3 Demucs	23
4.4 MusicExtractor	23
<b>5 Results</b>	<b>24</b>
5.1 MusiCNN and VGG	24
5.2 Spleeter and Demucs	27
5.3 MusicExtractor	34
<b>6 Audio visualisation libraries</b>	<b>38</b>
6.1 Matplotlib	38
6.2 Seaborn	39
6.3 Librosa	39
6.4 Bokeh	42
6.5 Conclusions	46
<b>7 Kandinsky's theories for colour and form</b>	<b>47</b>
7.1 Colour harmony	48
7.2 Form harmony	51
7.3 Choice of object	52
7.4 Composition	52
<b>8 Final composition</b>	<b>53</b>
8.1 Background	53

8.2	Forms . . . . .	55
8.3	Colours . . . . .	58
8.4	Composition . . . . .	58
<b>9</b>	<b>Conclusions . . . . .</b>	<b>61</b>
	<b>References . . . . .</b>	<b>65</b>
	<b>Appendixes . . . . .</b>	<b>69</b>

## FIGURES

Figure 1.	Google Arts Culture project named "Play a Kandinsky". The painting Yellow-Red-Blue can be seen in the background, while clickable elements appear on top of it, to be interacted with. (Google Arts & Culture 2022) . . . . .	13
Figure 2.	Visualisation of research process. . . . .	19
Figure 3.	Waveforms of the sound samples . . . . .	20
Figure 4.	Example of the MusiCNN model MSD dataset autotagging for Audio 1	26
Figure 5.	Example of the MusiCNN model MTT dataset autotagging for Audio 1	26
Figure 6.	Audio 1 MusiCNN MTT labels corresponding to separated sources. . .	29
Figure 7.	Audio 2 MusiCNN MTT labels corresponding to separated sources. . .	32
Figure 8.	Audio 3 MusiCNN MTT labels corresponding to separated sources. . .	33
Figure 9.	Audio 1 comparison of source files when computing audio features with MusicExtractor . . . . .	36
Figure 10.	Audio 2 comparison of source files when computing audio features with MusicExtractor . . . . .	36
Figure 11.	Audio 3 comparison of source files when computing audio features with MusicExtractor . . . . .	36
Figure 12.	Matplotlib sample audio waveform. . . . .	38
Figure 13.	Matplotlib spectrogram visualisation of an audio. . . . .	39
Figure 14.	Seaborn replotting function used to plot data. . . . .	40
Figure 15.	Seaborn replotting function used to replot the data seen in Figure 14 into different graphical plots . . . . .	40
Figure 16.	Librosa's waveplot graph comparing waveforms in mono, stereo, harmonic and percussive signals. . . . .	41
Figure 17.	Librosa's Short-Time Fourier Transform graph . . . . .	41
Figure 18.	Librosa's specshow graph . . . . .	42
Figure 19.	Example of Bokeh ridgeplot . . . . .	43
Figure 20.	Example of Bokeh hexbin plot . . . . .	44
Figure 21.	Example of Bokeh hextile plot . . . . .	44
Figure 22.	Example of Bokeh RGBA plot . . . . .	45

Figure 23. Bokeh patch plot . . . . .	45
Figure 24. Yellow-Red-Blue by Wassily Kandinsky (Wassily-Kandinsky 2022) . . . . .	47
Figure 25. Waveform visualisation of My Type by Saint motel . . . . .	53
Figure 26. Top VGG MSD mood labels . . . . .	53
Figure 27. Top VGG MTT mood labels . . . . .	53
Figure 28. Background for the audio visualisation . . . . .	55
Figure 29. Separated instrumental sources for the audio. . . . .	55
Figure 30. Instrumental sources dynamic complexity and dissonance. . . . .	57
Figure 31. Final visual composition of the song My Type . . . . .	60
Figure 32. MusiCNN MSD, audio 1 . . . . .	69
Figure 33. MusiCNN MSD, audio 2 . . . . .	70
Figure 34. MusiCNN MSD, audio 3 . . . . .	70
Figure 35. MusiCNN MTT, audio 1 . . . . .	71
Figure 36. MusiCNN MTT, audio 2 . . . . .	71
Figure 37. MusiCNN MTT, audio 3 . . . . .	72
Figure 38. VGG MSD, audio 1 . . . . .	72
Figure 39. VGG MSD, audio 2 . . . . .	73
Figure 40. VGG MSD, audio 3 . . . . .	73
Figure 41. VGG MTT, audio 1 . . . . .	74
Figure 42. VGG MTT, audio 2 . . . . .	74
Figure 43. VGG MTT, audio 3 . . . . .	75

## TABLES

Table 1.	MusiCNN model performance between MSD and MTT training datasets	24
Table 2.	VGG model performance between MSD and MTT training datasets	24
Table 3.	Audio 1: Spleeter and Demucs source separation comparison by waveforms	28
Table 4.	Audio 1: Spleeter and Demucs source separation comparison by mel-spectrograms	30
Table 5.	Audio 2: Spleeter and Demucs source separation comparison by waveforms	31
Table 6.	Audio 3: Spleeter and Demucs source separation comparison by waveforms	32
Table 7.	Audio features extracted for the whole audio and separate sources with MusicExtractor.	56
Table 8.	Forms for different instrumental sources.	57
Table 9.	Colours for different instrumental sources.	58

## DEFINITIONS

Before going to the actual research, we must understand some basics of musical theory. These are some definitions to keep in mind when selecting components for visualising musical audio data. In order to understand musical data, we must first understand what music is and what it's made of.

The work of Harry F. Olson, a pioneer in sound engineering, is an excellent source for information about the foundations of music and his book *Music, Physics, and Engineering* is used in this research as the guide to the building blocks of music.

Olson describes music to be "the art of producing pleasing, expressive, or intelligible combinations of tones" (Olson 1967). This can be understood in a way that music itself is already art and even though the musical audio data can be viewed as facts, they represent an artistic entirety.

Olson also breaks sound first into various physical properties and then into several psychological characteristics that are dependent on the afore mentioned physical properties. These are described as:

- Physical properties: frequency, intensity, waveform, and time. (Olson 1967)
- Psychological characteristics: pitch, loudness, time, and timbre. (Olson 1967)

## Sound

Sound itself can be described as a "mechanical radiant energy that is transmitted by longitudinal pressure waves in a material medium" (Merriam-Webster 2021c).

Shortly put, sound is energy which travels by sound waves. This means that when sound is received by the listener, it is transmitted via the medium of sound waves. However, it's good to note, that sound energy can be transmitted even though an audible sound cannot be heard by a human ear. (Merriam-Webster 2021c)

Olson defines sound waves in terms of six physical variables: frequency, intensity, wave-



form, duration, growth and decay, and vibrato (Olson 1967).

## **Frequency**

Olson defines the frequency to be "the number of cycles occurring per unit of time. It's determined by counting waves per second. A complete set of recurrent values of a periodic quantity comprises a cycle." (Olson 1967)

## **Intensity**

Olson defines the intensity of a sound to be "the sound energy transmitted per unit of time in the specified direction through a unit area normal to this direction at the point." (Olson 1967)

## **Amplitude**

The definition for amplitude according to the encyclopedia Britannica is the maximum displacement of a sound from its position of balance, or equilibrium. "The equilibrium value of pressure is equal to the atmospheric pressure that would prevail in the absence of the sound wave." (Britannica 2021)

Britannica also defines that "a fluctuation above and below atmospheric pressure occurs in a sound wave. The magnitude of this fluctuation from equilibrium is known as the amplitude of the sound wave." (Britannica 2021)

## **Decibel scale**

So far, we have defined that the amplitude of a sound wave determines its intensity, which in turn is perceived by the ear as loudness. According to Britannica, the human ear responds much more efficiently to sounds of very small amplitude than to sounds of very large amplitude, and this behaviour is called nonlinear. Because of this human ear mechanism, also a nonlinear scale is convenient when describing the intensity of sound waves. This scale is provided by the sound intensity level, or decibel level, of a sound wave. (Britannica 2021)

## **Waveform**

Olson defines that "a sound wave is made up of sound frequency and overtones. The overtone or harmonic structure is expressed in the number, intensity, distribution and phase relations of the components". (Olson 1967)

## **Duration**

According to Olson, the "duration of a sound is the length of time that a tone persists or lasts. The unit is the second or some submultiple of a second". (Olson 1967)

## **Pitch**

The definition for pitch is very interesting. Olson describes pitch to be "a sensory characteristic arising out of frequency, which may assign to a tone a position in a musical scale. The lower limit of pitch is the lowest frequency, which gives us a sensation of tone. The lower limit depends upon the individual and a number of physical factors, such as the intensity and the waveform of the sound". (Olson 1967)

This means that the pitch can refer to a perceived highness or lowness, relating to the frequency of the sound. Higher the frequency, higher the perceived pitch.

## **Tempo**

Tempo is a term used to identify the rate of movement or non-movement, the pauses and moments in-between, in the music. (Olson 1967)

## **Timbre**

The definition for timbre is a bit elusive. However, according to Isabella Van Elferen it's one of the most obscure areas of music and sound research. She defines the timbre to be "the most basic attribute not just of music, but of sound: there is no sound without timbre, and sound does not even need to be heard to have a timbre". (Van Elferen 2020)

What makes timbre at the same time so abstract and important in musical audio analysis, is its "quality of tone distinctive of a particular singing voice or musical instrument" (Merriam-Webster 2021d), which means it can be used to separate one instrument from

another and be one of the main elements for an audio classification.

## **Tone**

The tone of a musical instrument consists of a frequency and several harmonics and partials. Depending on the type of instrument, the tone is characterized by the attack and decay. Each partial may have different attack and decay patterns. (Brixen 2020)

## **Dynamic complexity**

The dynamic complexity is an algorithm by python library Essentia, and it defines "the average absolute deviation from the global loudness level estimate on the dB scale. It's also related to the dynamic range and to the amount of fluctuation in loudness present in a recording". (Essentia 2022b)

## **Dissonance**

The standard deviation of dissonance is also an algorithm by Essentia, and it's calculated by the "sensory dissonance of an audio signal given its spectral peaks". There are different forms of dissonance, but the sensory dissonance "measures perceptual roughness of the sound and is based on the roughness of its spectral peaks." (Essentia 2022a)

## FOREWORD

*“Colour is a power which directly influences the soul. Colour is the keyboard, the eyes are the hammer, the soul is the piano with many strings. The artist is the hand which plays, touching one key or another, to cause vibrations in the soul.”*

— *Wassily Kandinsky*

There’s always been people who can visualise music in their minds, but it’s unclear why the shape and colour of a sound come so easily to others. Some people paint waves and points and splashes into the air with their hands, based on the rhythm, melody, and harmony of the music. They might not be able to play any instrument, but still hold close the love for music, and in addition to hearing music, they are able to see it.

It has always been a close interest to me because I can visualise music in my mind and as a movement in my body. However, not being a musician or an artist myself, the curiosity of this phenomenon has just been sitting at the back of my mind.

The idea for this thesis, to combine data with art, started from stumbling into the Google Arts & Culture experiment of Wassily Kandinsky’s painting *Yellow-Red-Blue*. It gave a spark to wonder how Kandinsky’s supposed process of turning a sound he heard, into visual art masterpieces, could be recreated with machine learning methods.

I want to thank my supervisor Leonardo Espinosa-Leal for patiently brainstorming the topics of my thesis to make sense out of random ideas and for continuously guiding and pushing me towards more meaningful research. This thesis is a starting point into my journey towards data art.

Helsinki, April 2022

Jaana Moilanen

# 1 INTRODUCTION

The inspiration for this research comes from the abstract painter Wassily Kandinsky, who was said to have a neurological condition called sound synaesthesia (Denver Art Museum 2021). This thesis aims to understand how it would be possible to use musical data to create visual art. We've researched ways to use music, and art theory combined with machine learning methods to create artistic visualisations of musical audio data.

Musical theory helps to understand how music is constructed, how different instruments help create emotions and spark imagination, for example with different tones, rhythms, pitches, and timbres. When visualising musical data, the outcome should provide representations for feelings music has created in the listener. The interpretation of those feelings is the art itself.

The use of data in all aspects of life is becoming more and more advanced. There are algorithms used in advancing human life such as trading algorithms, in modifying human behaviour every day such as internet algorithms, and protecting all this data such as encryption algorithms. But even after all of these advances, it's immensely more difficult and relates back to defining the terms "data" and "art".

The Merriam-Webster dictionary defines data to be "facts or information usually used to calculate, analyse, or plan" (Merriam-Webster 2021b), and defines art to be "something that is created with imagination and skill and that is beautiful or that expresses important ideas or feelings" (Merriam-Webster 2021a).

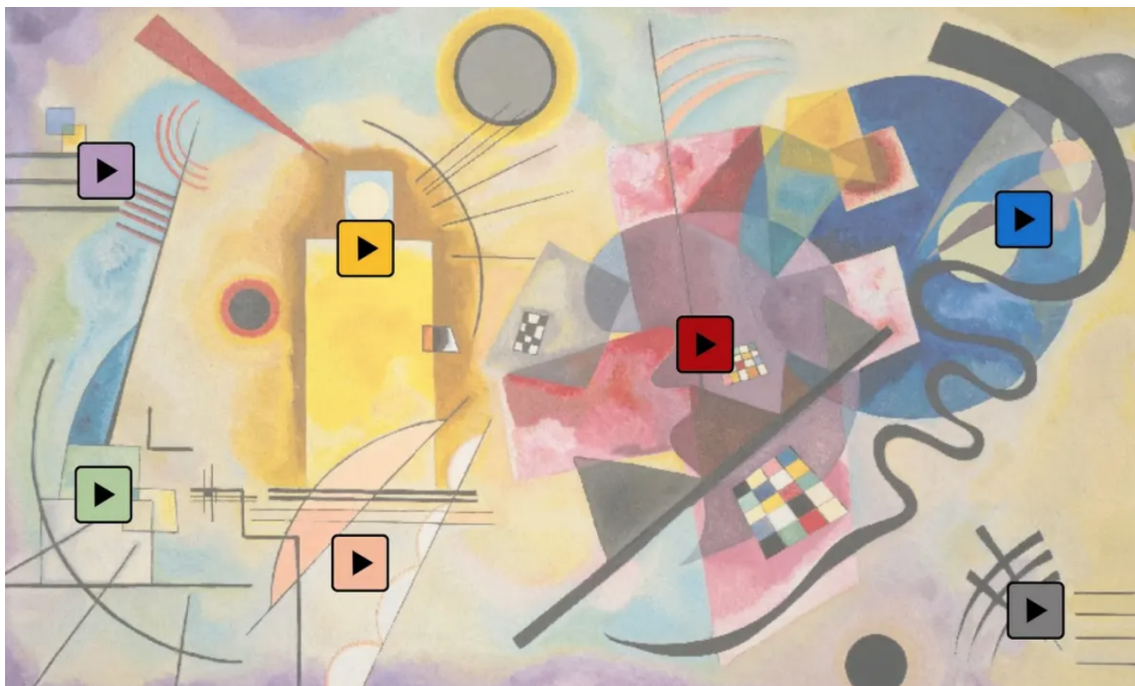
This leads to a discrepancy between facts and imagination. How can we use facts from musical audio data to create something imaginative and beautiful which expresses ideas or feelings? We need to understand how to use data to create. And in this research we're studying how to use musical audio data to create not musical art, but visual art.

In a research done by K. Itoh, H. Sakata, H. Igarashi and T. Nakada reviewing synaesthesia in musicians there were two examples of how sound creates an association with brightness and colour. These two forms of synaesthesia, induced by sound pitches, are called pitch height colour synaesthesia and pitch class colour synaesthesia. (Itoh et al.

2019)

- **Pitch height-colour synaesthesia** is the cross modal association between pitch height and lightness/brightness. (Itoh et al. 2019)
- In **pitch class-colour synaesthesia**, the higher-order concept of pitch class, rather than the sensory-level highness/lowness of a tone, induces the concurrent sensations of colour. (Itoh et al. 2019)

In Kandinsky's case, when he heard a sound it triggered a colour in his vision and when he painted and saw colours it triggered sounds in his ears. This phenomenon has interested researchers through the years to the extent that Google has created a "Play a Kandinsky" -experiment (Google Arts & Culture 2022), seen in Figure 1, where a user can play Kandinsky's Yellow-Red-Blue painting and hear sounds with each colour and shape.



*Figure 1. Google Arts Culture project named "Play a Kandinsky". The painting Yellow-Red-Blue can be seen in the background, while clickable elements appear on top of it, to be interacted with. (Google Arts & Culture 2022)*

Google worked with musicians to study Kandinsky's writings detailing his multisensory capabilities, and with machine learning experts to simulate what Kandinsky might have heard as the Yellow-Red-Blue -painting was being created.

## 2 RELATED WORK

This thesis aims to study if we can use machine learning to create visualisations of musical data, similarly as some people might see music as colours in sound synaesthesia. There are two sides to this topic: first to understand the music theory to utilise in machine learning algorithms and how to visualise this data as art in ways that mimic the sound synaesthesia Kandinsky said to have had.

Here are some research that focuses on understanding different models and algorithms done with musical audio data.

### 2.1 Structural analysis of recorded music

In a very recent research, Lauri Saikkonen studied the structural analysis of music in one subfield of music information retrieval in his thesis of Structural analysis of recorded music, which he did in collaboration with the Yousician application. His research provided many good starting points for this thesis.

The objective of the study was to take a look into some state-of-the-art methods for music structure analysis and find out if automatic chord transcription can be improved using these methods. It examined different methods and outlined several models for chord transcription improvement and to create structural analysis of recorded music. (Saikkonen et al. 2020)

The study states that one of the most essential constructions in the task of music structure analysis is the self-similarity matrix (SSM). The self-similarity matrix describes how similar each point of time in a recording is to another point of time in that same recording. The acoustic features and the similarity measure define how the SSM behaves and affects how relevant information can be retrieved from it. (Saikkonen et al. 2020)

The study researched the Mauch et al. method by analysing a self-similarity matrix which uses beatsynchronous chromagram as features. First, they constructed a chromagram, where the median chroma values were then taken from the frames and then the beats were calculated automatically. Another method they used was Serra et al. which analyses

structural boundaries and detects repeating sections and is also based on analysing the self-similarity matrix. (Saikkonen et al. 2020)

The research also used a method of Ullrich, Schluter and Grill which calculates a novelty curve using different representations of audio as an input to a convolutional neural network (CNN). Then, in post-processing, the novelty curve is analysed and the peaks from it are picked. Interestingly, the research noted that spectrograms can be padded with pink noise, to make boundary detection better at the edges of the audio. (Saikkonen et al. 2020)

Saikkonen also found out that the methods didn't take into account any temporal deviations which naturally would occur in music. One method of matching time series and allowing matching of sequences which have local temporal deviations is dynamic time warping (DTW). With the DTW algorithm, one can calculate the optimal warping path between two sequences. (Saikkonen et al. 2020)

Also, it was declared that, especially in the task of finding sectional boundaries, it's of utmost importance that the timing of the annotation matches exactly that of the audio. Thus, only a part of the dataset, which had links to the original audio, was used. (Saikkonen et al. 2020)

## **2.2 Investigating audio data visualization for interactive sound recognition**

Research by T. Ishibashi, Y. Nakao, and Y. Sugano, was focused on investigating audio data visualization for interactive sound recognition. Their work studied the design issues for interactive sound recognition by comparing different visualization techniques, ranging from audio spectrograms to deep learning-based audio-to-image retrieval. Based on an analysis of user study, it clarifies the advantages and disadvantages of audio visualization techniques, and provides design implications for interactive sound recognition with graphical user interfaces (GUIs) using a massive amount of audio samples. (Ishibashi et al. 2020)



Interactive sound recognition is not clearly related to the research questions of this thesis, but the design implications for interactive sound recognition in GUIs are an interesting research area, which are to be considered.

### **2.3 A Survey on Visualizations for Musical Data**

A Survey on Visualizations for Musical Data researched visualizations available for musical data and focused on the link between musicology and visualization. (Khulusi et al. 2020)

The study classifies 129 related works according to the visualized data types, and analyses which visualization techniques were applied for certain research inquiries and to fulfil specific tasks. Next to scientific references, it takes commercial music software and public websites into account that contribute novel concepts of visualizing musicological data. (Khulusi et al. 2020)

### **2.4 Audio Event Classification Using Deep Learning Methods**

In research of Audio Event Classification Using Deep Learning Methods, Zhicun Xu studied audio event classification using deep learning methods and focused on studying deep learning methods and building suitable neural networks for assigning labels to particular audio signals. (Xu et al. 2018)

The evaluation of the performance of different neural networks is tested on both Google AudioSet and the dataset for DCASE 2018 Task 2. This study would have come in handy if we'd decided to train an algorithm with Google AudioSet to be used in this thesis.

### **2.5 Visualization in Audio-Based Music Information Retrieval**

An article Visualization in Audio-Based Music Information Retrieval introduced different visualization techniques developed in the context of Music Information Retrieval (MIR) for representing polyphonic audio signals. (Cooper et al. 2006)

Although this article was written 14 years ago, it still provides some valuable information about MIR algorithms to be used in this thesis. The article focuses on Music Information Retrieval, which is a research area that explores how music stored digitally can be effectively organized, searched, retrieved and browsed.

In the article, they utilised analysis algorithms to automatically extract content information from music pieces stored in audio format. There seems to be some benefits in their research and techniques to be implemented in this thesis.

## **2.6 Hybrid Spectrogram and Waveform Source Separation**

Very recent research of Hybrid Spectrogram and Waveform Source Separation by Alexandre Défossez from Facebook AI department, displays modern end-to-end hybrid audio source separation on the spectrogram or waveform domain, called Demucs Music Source Separation model. (Défossez 2021)

The research extends the original architectures to provide temporal and spectral branches to perform hybrid waveform source separation for audio into instrumental segments, and outputs raw waveform for each of the source; drums, bass, vocals and other accompaniments.

The Demucs Music Source Separation model proved to be a very useful way to separate instrumental sources from music, as will be presented in this thesis.

### **3 RESEARCH METHODOLOGY**

The first phase of this research process is to classify the overall musical genre and mood for the selected song, which are then combined with Kandinsky's colour definitions to create the visualisation for the background and general forms in the visualisation.

After mood and genre classification, the audio is separated into instrumental sources. These sources are then used to extract specific audio features, such as pitch, tone, tempo, etc. for each instrument. The entire audio is also used to extract its overall audio features.

The features are then combined with Kandinsky's colour and form definitions, which give us enough information to visualise each instrument with its unique shape and colour. Finally, we'll combine all individual forms together to create a one big Kandinsky.

#### **3.1 Audio classification**

The audio classification can be done either by training a new model or using an existing, pre-trained, model available. Before going to train a new model, we wanted to know if it's enough to use a pre-trained algorithm to classify audio for us. We compared MusicCNN and VGG models for audio labelling and MusicExtractor for extracting audio features, all from Essentia's audio analysis library.

For source separation, we compared Spleeter and Demucs models, from which Spleeter is capable of separating the audio up to five sources, and Demucs model into four.

The whole process is described in the Figure 2. The letters next to MusicExtractor features (i.e., "Pitch (B)") indicate the RGB colour values the specific feature will affect, R for amount of red, G for amount of green and B for amount of blue.

#### **3.2 Audio visualization libraries**

To be able to understand what kind of visualizations are currently possible, we've compared some generic audio data visualization methods available in the python libraries, such as Matplotlib, Seaborn, Plotly, Bokeh and Librosa. After comparison and testing,

we've selected libraries that will give us flexible ways of visualising forms with different colours.

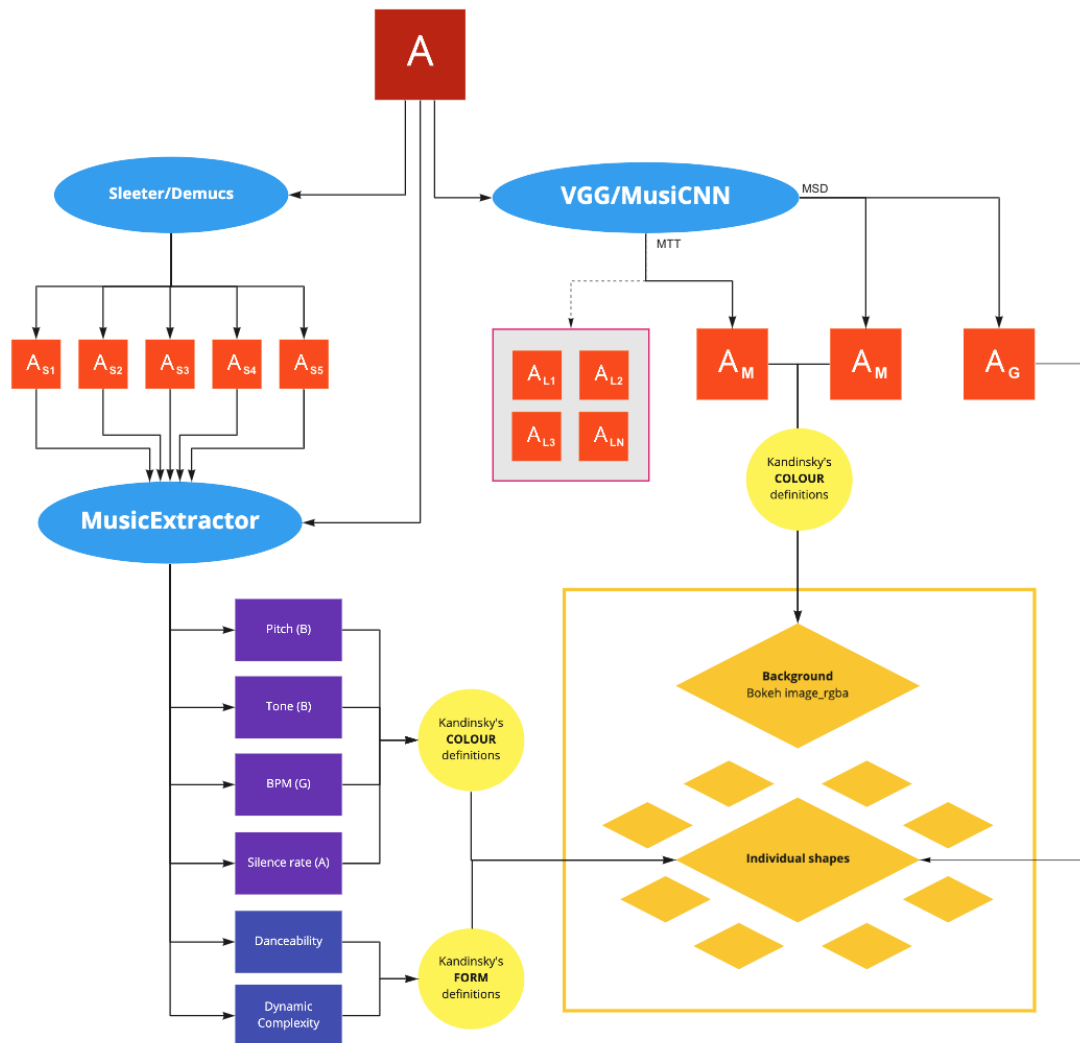


Figure 2. Visualisation of research process.

### 3.3 Visualization of musical audio data

The aim for this research is to create a holistic visualization of musical audio in a way that mimics what a person with sound synaesthesia would see, as they hear the musical audio. The visualisation could help the viewer to understand what Kandinsky could have imagined while he was painting his masterpiece Yellow-Red-Blue.

It will not be an independent piece of art, but a part of a simulation of what the process of creating Yellow-Red-Blue might've been like for him. The journey, which will start from a piece of music, will continue to define colours and shapes and ends in a combination of

understanding how music is heard, through the viewer's eyes.

### Audio sample details

The process is first tested and optimised on three different musical audio from three different genres. The same three audio files are used when testing machine learning models, source separation and feature extraction. However, once the process is completed, its optimised version is used to create colours and shapes for the final visualisation of a separate, fourth, musical audio file. Details for this is provided later.

Details for three audio that are used in testing of the process are:

- Audio 1: **Feather** by Ma Rouf, categorised as modern fusion jazz (duration 4:47) (MA Rouf 2019)
- Audio 2: **Vivaldi's Winter Mvt 1 Allegro non molto** by John Harrison with the Wichita State University Chamber Players, categorised as classical and baroque (duration 3:29) (Vivaldi, A. 1723)
- Audio 3: **Upbeat Forever** by Kevin MacLeod, categorised as rock (duration 3:15) (MacLeod, K. 2022)

These songs are similar in a way, that in none of them there are any vocals present, but they are from different genre and have difference i.e., in the waveform, as seen in Figure 3.

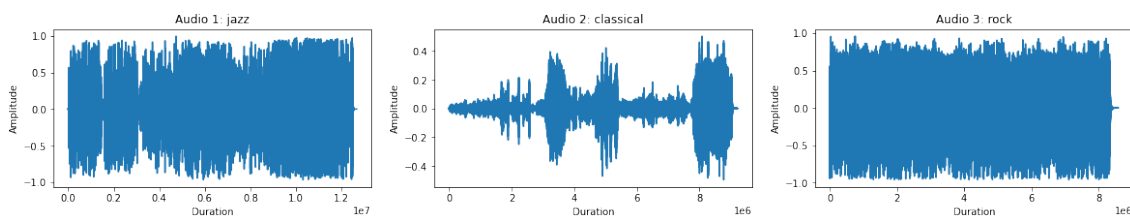


Figure 3. Waveforms of the sound samples

## 4 EXPERIMENTS

### 4.1 MusiCNN and VGG

Essentia is a vast library specified for audio and music analysis, description and synthesis. It's designed to offer the flexibility of use, easy extensibility, and real-time inference, and allows using TensorFlow models within an audio analysis framework. (Essentia 2021)

We're testing two models from Essentia: MusiCNN and VGG, both trained with two different datasets, MSD and MTT. These are tested with three different genres of musical audio files: jazz, classical and rock. Both models include auto-tagging, which is going to be used in the audio data classification process.

There is an option to use multiple MTG in-house datasets, which are a collection of smaller and highly curated datasets used for training classifiers (Essentia Labs 2021). We chose to focus on the MSD and MTT models as they provide 50 top tags, which is much more than any of the smaller datasets.

#### Architecture details

- **MusiCNN** model is a musically-motivated Convolutional Neural Network. It uses vertical and horizontal convolutional filters aiming to capture timbral and temporal patterns, respectively. The model contains 6 layers and 790k parameters. (Essentia Labs 2021)
- **VGG** model is an architecture from computer vision based on a deep stack of commonly used 3x3 convolutional filters. It contains 5 layers with 128 filters each. Batch normalization and dropout are applied before each layer. The model has 605k trainable parameters. (Essentia Labs 2021)

MusiCNN and VGG models are used for genre and mood classification. They are also used in estimating the acoustic and instrumental labels for the sample songs. Getting an accurate result is not necessary, as the classification will only be serving as an indication of what the source separation of the audio files should be taking into consideration.

## Dataset details

There were two different datasets used in training the models; the MSD (Million Song Dataset) and MTT (MagnaTagATune) datasets. The comparisons are run by using the default parameters to see how fast and accurately the model performs in the classification process.

- **MSD** contains 200k tracks from the train set of the publicly available Million Song Dataset, annotated by tags from Last.fm. Only the top 50 tags are used. (Essentia Labs 2021)
- **MTT** contains 25k tracks from Magnatune with tags by human annotators. Only the top 50 tags are used. (Essentia Labs 2021)

**MSD** dataset is based on 50 different labels, which are a combination of genre and mood descriptors: rock, pop, alternative, indie, electronic, female vocalists, dance, 00s, alternative rock, jazz, beautiful, metal, chillout, male vocalists, classic rock, soul, indie rock, Mellow, electronica, 80s, folk, 90s, chill, instrumental, punk, oldies, blues, hard rock, ambient, acoustic, experimental, female vocalist, guitar, Hip-Hop, 70s, party, country, easy listening, sexy, catchy, funk, electro, heavy metal, Progressive rock, 60s, rnb, indie pop, sad, House, and happy.

**MTT** dataset is based on labels more focused to instrumental descriptors, but some mood and genre tags are also found: guitar, classical, slow, techno, strings, drums, electronic, rock, fast, piano, ambient, beat, violin, vocal, synth, female, indian, opera, male, singing, vocals, no vocals, harpsichord, loud, quiet, flute, woman, male vocal, no vocal, pop, soft, sitar, solo, man, classic, choir, voice, new age, dance, male voice, female vocal, beats, harp, cello, no voice, weird, country, metal, female voice, and choral.

## 4.2 Spleeter

Spleeter is a source separation library, from the music streaming platform Deezer, with pretrained models written in Python and uses Tensorflow. It makes it easy to train the source separation model, and provides an already trained state-of-the-art model for per-

forming various separation styles. (Deezer 2022)

Spleeter has a ready "Spleeter" model that works with Python, which is used in the source separation process for the audio files.

### **4.3 Demucs**

Demucs Music Source Separation model by Alexandre Défossez focuses on using only the spectral and temporal data in the classification process. This work is interesting as it provides source separation on with a pre-trained model and easy-to-use python bindings.

The model is trained partly with musdb18 is a dataset which of 150 full lengths' music tracks (approx. 10h duration) of different genres along with their isolated drums, bass, vocals, and others stems. (SigSep 2021)

The Demucs source separation model is compared side-by-side with the Spleeter, to see which model would give more performance and accurate results.

### **4.4 MusicExtractor**

Essentia provides a configurable command-line feature extractor that computes a large set of spectral, time-domain, rhythm, tonal and high-level descriptors, called the MusicExtractor. (Essentia 2021)

The MusicExtractor is able to calculate different features of the audio data, which can be used as parameters when creating visualisations.

These audio features include i.e., replay gain, average loudness, integrated loudness, loudness range, silence rate, HFC mean, Pitch salience, zero crossing rate, tempo or BPM, Beats loudness, danceability score, onset rate, chords change rate, chords key, key/scale estimations, dynamic complexity, spectral complexity and energy, and many more.

In this research, we're mainly interested in features estimating i.e., pitch, tone, danceability, tempo, and audio complexity.



## 5 RESULTS

### 5.1 MusiCNN and VGG

Comparing the results of MusiCNN auto-tagging for all three sample songs was done by creating a spectrogram analysis and estimation for most prominent tags. Tags represent either the genres (MSD) or also instrumental characteristics (MTT) for the song, depending on the training dataset used.

We've compared the Top 5 tags the model categorised the song to belong into, and the amount of time spent in the estimation, as it might critically affect the speed of the model.

*Table 1. MusiCNN model performance between MSD and MTT training datasets*

AUDIO	DATASET	EST. TIME	TAGS
Audio 1	MSD	7.57 seconds	jazz, instrumental, funk, blues, rock
Audio 1	MTT	8.23 seconds	guitar, indian, drums, strings, sitar
Audio 2	MSD	6.01 seconds	instrumental, jazz, folk, rock, progressive rock
Audio 2	MTT	6.10 seconds	classical, violin, strings, classic, fast
Audio 3	MSD	5.49 seconds	rock, jazz, instrumental, indie, alternative
Audio 3	MTT	6.35 seconds	drums, guitar, rock, beat, pop

Using the same pipeline, but changing the model from MusiCNN to VGG gives us fairly similar results. Top 5 tags and time for estimation using the VGG model are as follows:

*Table 2. VGG model performance between MSD and MTT training datasets*

AUDIO	DATASET	EST. TIME	TAGS
Audio 1	MSD	6.49 seconds	jazz, electronic, rock, instrumental, indie
Audio 1	MTT	5.79 seconds	guitar, drums, indian, fast, strings
Audio 2	MSD	4.65 seconds	jazz, rock, electronic, instrumental, folk
Audio 2	MTT	4.49 seconds	classical, violin, strings, opera, classic
Audio 3	MSD	4.83 seconds	rock, indie, alternative, electronic, experimental
Audio 3	MTT	4.27 seconds	guitar, drums, rock fast, beat

VGG model comparison was done in the similar way as MusiCNN. Auto-tagging was created by calculating a spectrogram analysis and estimation for most predominant tags.

Tags also represent either the genres (MSD) or also instrumental characteristics (MTT) for the song.

## Results

The VGG model itself seems to mostly affect the time it takes to estimate results, where VGG is performing slightly faster than MusiCNN. However, changing the dataset used for training, gives very different auto-tagging results. This is because the classification labels in MSD and MTT datasets are different. Whereas the labels in MSD dataset focus more in the genre and mood of the audio, the MTT dataset has been labelled both with genre and instrumental descriptors, and it's more focused on the instruments.

Comparing these results from the two models, we can see that MSD was not able to properly categorise Audio 2 'Vivaldi's Winter' as classical music. Further inspection of the tagging used in the dataset, explains that this is because it doesn't contain a label for "classical" music.

The MTT dataset trained models automatically classify instruments and genres, as well as many similar tags. The middle-eastern sitar-like instrument, used in Audio 1, is classified to most probably be a guitar. References to the same instrument are in multiple of the most predominant tags, as it's tagged as "guitar", "strings" and "sitar". MTT also classifies Audio 1 to belong into a genre "indian" which is not really accurate as the melody is Middle-Eastern origin, not Indian.

It's to be noted, that Essentia's models are creating a spectrogram by reproducing the training features that were computed with Librosa (Essentia Labs 2021).

An example of a spectral visualisation for auto-tagging Audio 1 with MusiCNN MSD dataset, can be seen in the Figure 4. The spectral data shows the confidence of the 50 top tags labelled in MSD dataset on audio timeframe. These tags include mostly genre and mood classifiers. For Audio 1 the most prominent label is set to "jazz", which is highly accurate as Audio 1 genre is jazz music.

An example of a spectral visualisation for auto-tagging Audio 1 with MusiCNN MTT

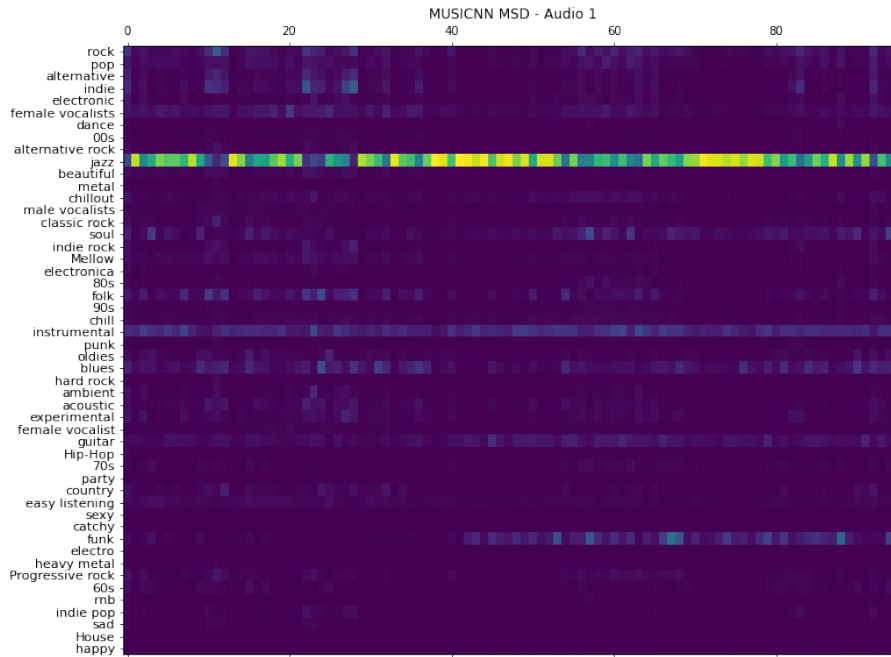


Figure 4. Example of the MusiCNN model MSD dataset autotagging for Audio 1

dataset can be seen in the Figure 5. The autotagged labels show significant differences to the MSD labels, as these tags include genre and instrumental classifiers. For Audio 1 the most prominent label is "guitar".

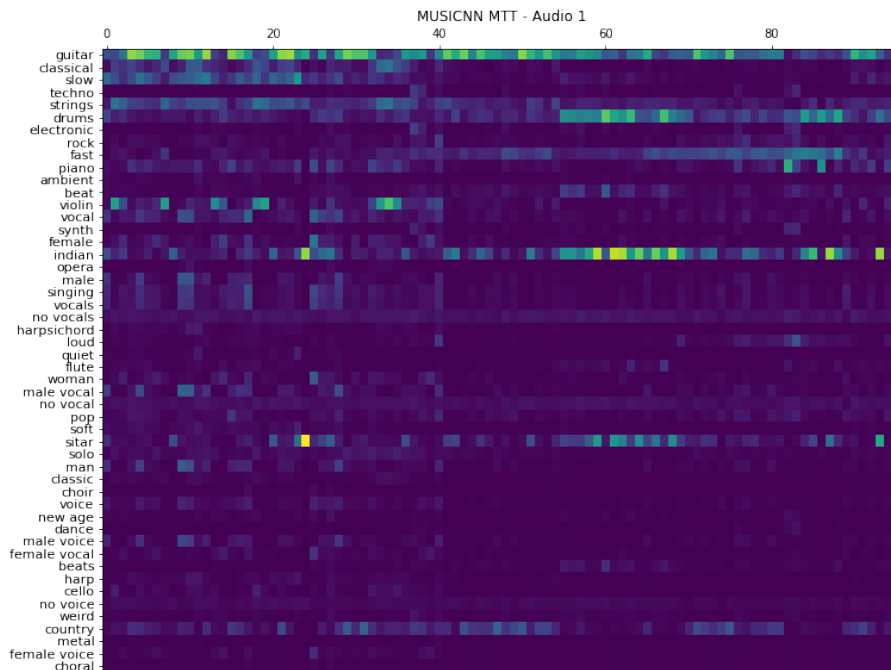


Figure 5. Example of the MusiCNN model MTT dataset autotagging for Audio 1

Spectral visualisations for both MusiCNN and VGG models with MTT and MSD datasets

for all audio files can be seen in the Appendixes 32 - 43.

## Conclusions

From the MusiCNN and VGG models, we can conclude even though they both produce very similar results, the VGG model works faster. However, we can benefit from both MSD and MTT dataset trained modelling.

From the MSD dataset, which categorises audio based on genre and mood, we can use the most prominent tag to give an overall mood for the visualisation. From the MTT tags, we can separate the most prominent instrumental tags to visualise more specific items.

For the visualisation we've selected to use the VGG model for its more fast estimation process and its selected descriptors either from the MSD or MTT datasets:

- The most prominent genre label (out of 50 genres)
- 11 instrumental labels
- 7 labels to depict the mood from the MTT dataset
- 7 labels to depict the mood from the MSD dataset

## 5.2 Spleeter and Demucs

The Spleeter model analyses the audio and separates five sources as wav-files, corresponding to labels: "drums", "bass", "vocals", "piano" and "others". The label "others" can us to contain all kinds of instrumental and vocal data that is not classified into the other sources for drums, bass, and vocals.

The Demucs model analyses the audio and separates four sources as wav-files, corresponding to labels: "drums", "bass", "vocals" and "others". As with Spleeter, the label "others" is used to contain all kinds of instrumental and vocal data that is not classified into the other sources for drums, bass, and vocals.

We've compared waveforms of the separated sources for each audio to see if there are

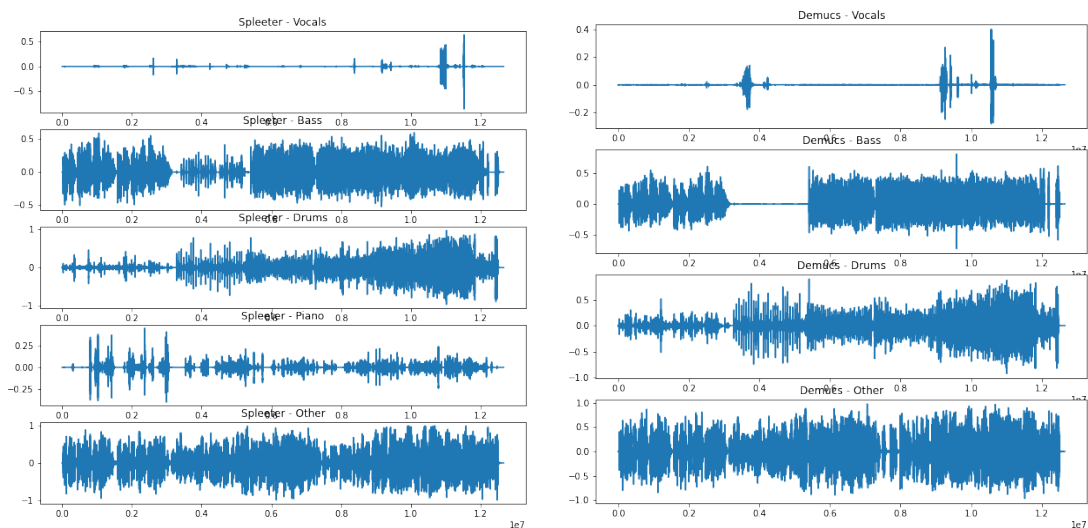
crucial difference between the results. We've also compared MusiCNN MTT tags for the instrumental sources, corresponding with the labels available.

When running the model, both of the models will try to separate instruments, even though they do not exist. For example, all three audio samples were given a separate source for "vocals" but there are no vocals present in any of them.

## AUDIO 1

The Audio 1 waveforms for the separated sources in Table 3 show how both models estimate there to be "vocals" even though there are no vocals present. We know this by listening to the audio, but the model perceives there to be some kind of vocal data.

Table 3. Audio 1: Spleeter and Demucs source separation comparison by waveforms



The Demucs model shows a clear absence of "bass" in a specific timestamp where Spleeter provides data for this source. It's unclear where this data is tagged in Demucs model, as the "drums" source seems to be similar in both models, as is the "other" source.

To be able to estimate the performance of the source separation, we reviewed the MusiCNN MTT dataset tags. The top tags for Audio 1 from the MusiCNN MTT dataset were "guitar", "indian", "drums", "strings" and "sitar", but these are not present in the separated sources.

For this, it was needed to filter all tags to correspond only those instruments present in Spleeter or Demucs model. However, it should be noted that these tags are not exactly the same as Spleeter sources ("vocals", "bass", "drums" and "piano") but tags that are corresponding to Spleeter sources as closely as possible.

From the Figure 6 we can see how MusiCNN has estimated there to be more "beat" or "beats" after halfway of the audio. We classify these corresponding to "bass", as there is a separate tag for "drums", which is very present. As do the Spleeter and Demucs waveforms, MusiCNN also estimates there to be lots of different types of vocals present.

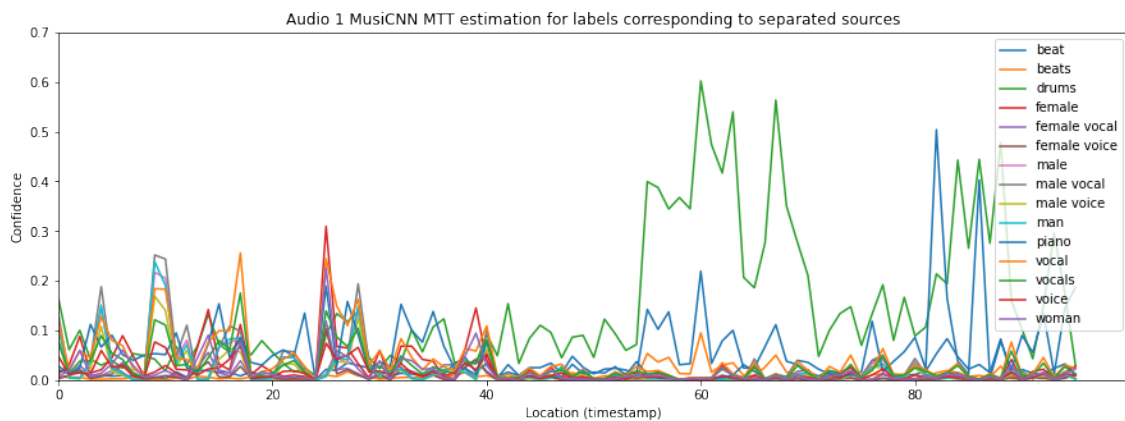
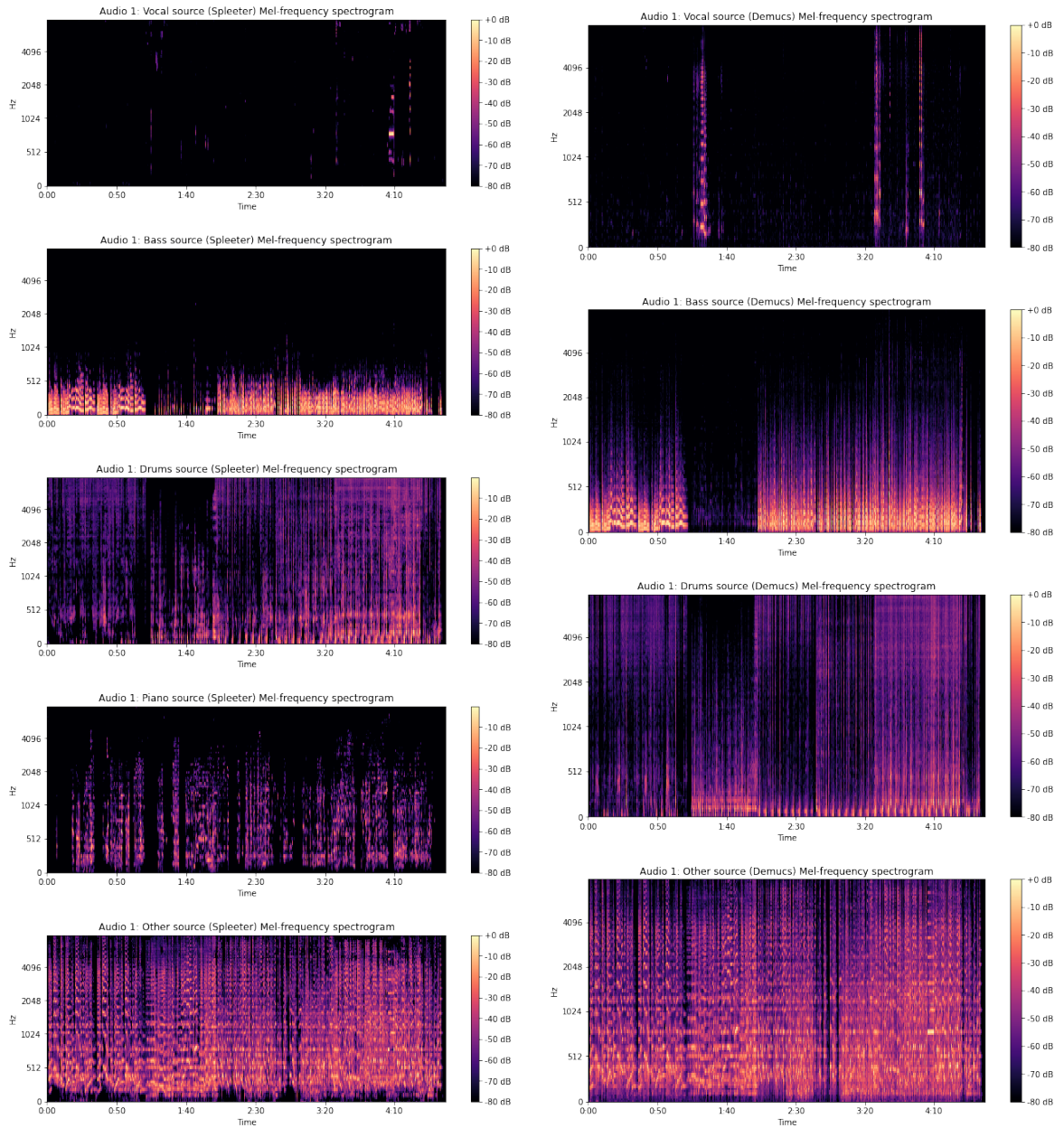


Figure 6. Audio 1 MusiCNN MTT labels corresponding to separated sources.

This causes some concerns on what kinds of music these source separation models can be used to gain most accurate data. However, it's to be kept in mind, that from the purposes of the final visualisation of musical data, we don't require the separated sources to be exact. It should be as close to accurate as possible, though.

Comparing the spectrograms for the separated sources of Audio 1 in Table 4, it's shown that Spleeter gives tighter amplitude levels for the audio than Demucs. However, we can't determine by this alone if the source separation would be more accurate in Spleeter.

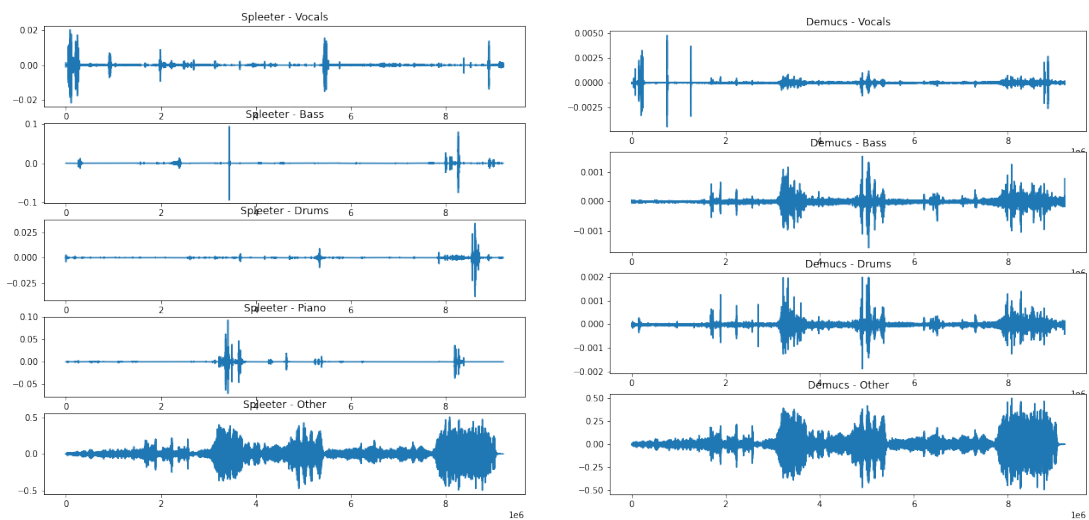
Table 4. Audio 1: Spleeter and Demucs source separation comparison by mel-spectrograms



## AUDIO 2

Audio 2 is categorized as classical music, and it seems to be very difficult for either model to separate its instrumental data accordingly. This seems to be especially hard for the Demucs model as, even though it appears to separate "bass" and "drums" on different sources, the waveforms in Table 5 show that they are almost identical. Spleeter gives only vague data for vocals, bass, drums, and piano. This raises the concerns if either of the models can be used with classical music source separation.

Table 5. Audio 2: Spleeter and Demucs source separation comparison by waveforms



Viewing the graph from the MusiCNN MTT label estimation in Figure 7, unfortunately gives little insights on what corresponding sources might be available. MusiCNN estimates there to be high amounts of "beat", "female" and "drums".

The appearance of vocal data, in the separated sources from Spleeter and Demucs, as well as in MusiCNN labels, again raises concerns that these models are not accurate enough to be used with all types of music.

It's to be noted, that there are other datasets available to be used with MusiCNN, but these did not provide as detailed information for the instrumental labels as the MTT dataset.



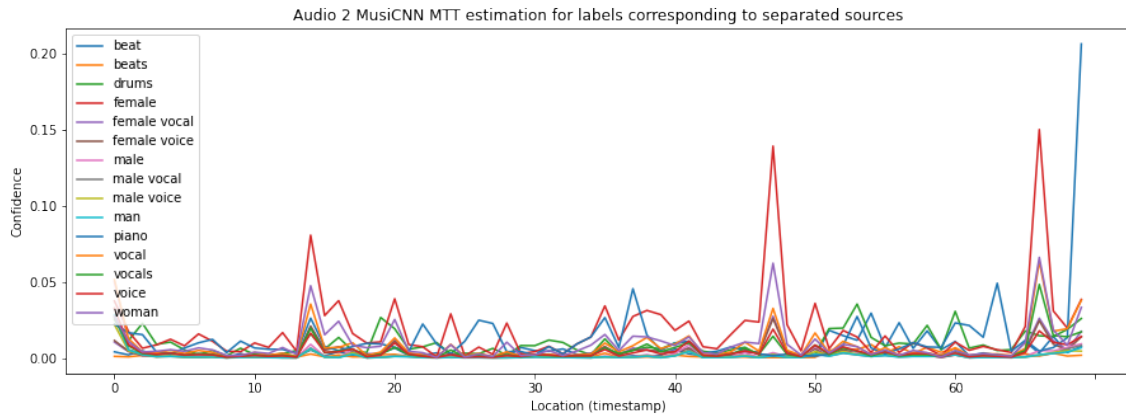
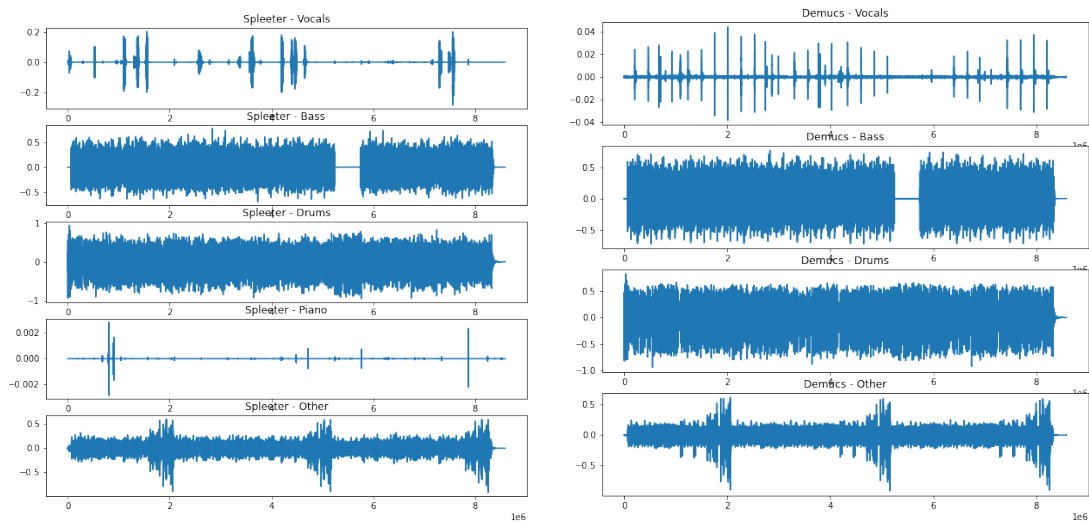


Figure 7. Audio 2 MusiCNN MTT labels corresponding to separated sources.

### AUDIO 3

Audio 3 is categorised as rock music, and it doesn't have vocals either. The source separation models are, however, estimating some "vocal" data for it as well, as can be seen comparing their waveforms in the Table 6. Other source waveforms seem to be very much alike in both models.

Table 6. Audio 3: Spleeter and Demucs source separation comparison by waveforms



Spleeter seems to find very small amounts of data for the "piano" source. There is no "piano" source in the Demucs model to compare the accuracy of "piano" source separation. However, we can view the MusiCNN MTT dataset labels in Figure 8 to see that MusiCNN also recognises some piano features. By listening to the audio, we can estimate that there

might be some electronic keyboard involved, which might be attributed into the "piano" source.

MusiCNN top tags for the Audio 3 are "beat", which we classify to indicate the presence of "bass", and "drums". There are some data for "piano" as well, and much of different types of vocal information. Again, this seems to be a fault in all the models.

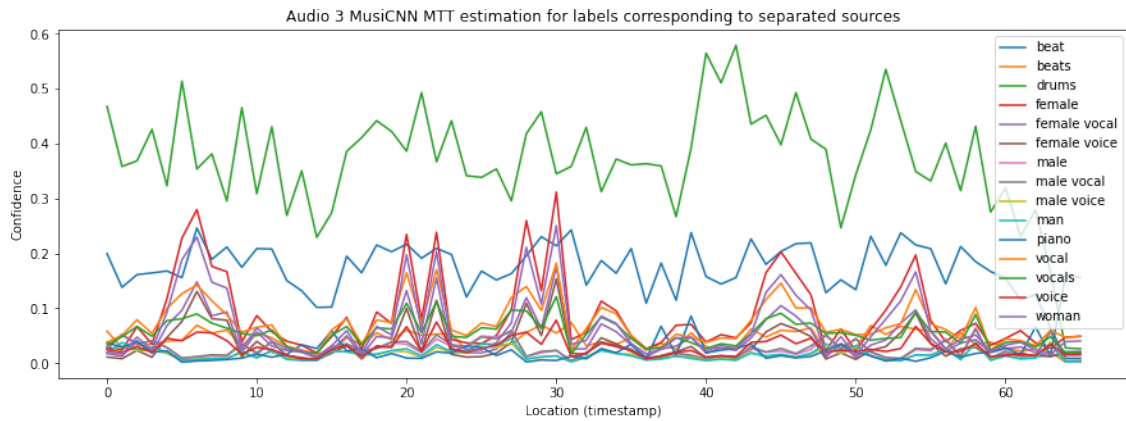


Figure 8. Audio 3 MusiCNN MTT labels corresponding to separated sources.

## Conclusions

When comparing Spleeter and Demucs, two pre-trained source separation models, We've had to keep in mind that in the final visualisation, it doesn't matter if the actual source data is exactly correct. As the final output will be visual art, it doesn't give explanation to the viewer which part of the audio is selected to be presented with a special colour or shape.

Also, when listening to the outputted source files, there are some differences between the models, but as that doesn't have relevance for the final visualisation as it doesn't use sound.

When comparing the results for source specific audio features, the two models are giving mostly similar results. The biggest differences seem to be in the data for vocal source, which doesn't actually exist in any of the audio samples.

However, there are some clear differences in performance, such as the duration it takes for the models to operate. Spleeter is significantly faster, doing separation even under a

minute, whereas Demucs has taken for the same audio ten times longer. For this difference in the Demucs model operating time, we would've expected higher performance.

Spleeter also has the advantage of 5 stem source separation as it also separates the source for "piano", which Demucs doesn't do. This is especially important in the type of audio which doesn't have vocals, bass, or drums used at all, as seen in Figure 10.

Demucs model could be used in of the visualisation process, but it currently doesn't give enough potential, compared to the Spleeter model, which gives more source separation and is also much faster. Keeping in mind the need for different instrumental visualisations, it's clear the Spleeter model will be more beneficial.

It's also to be noted, that these two models don't operate on their top performance if there isn't the correct sources to separate. All the audio samples have been music without vocals and some without bass or drums. It's evident that the source separation models work optimally with musical audio which contains data for all the possible separated sources, including vocals, as does MusiCNN model with MTT dataset.

For the final visualisation we've selected a song with all the possible separated instrumental sources are used; a pop song called *My Type*, performed by Saint Motel. Its music video can be seen on YouTube at <https://youtu.be/IyVPyKrx0Xo> (Saint Motel 2022).

### **5.3 MusicExtractor**

Essentia's MusicExtractor is an algorithm to compute different features from audio data. There are several features which can help transform the audio into a colour and shape.

The Silence rate (60d mean) could give an indication of how much sound there is at 60dB. We're interested in the silence rate of an audio because it could be used to calculate the amount of black or white in the visualisation.

The pitch salience mean is also an interesting tool to describe the pitch of an audio, and BPM to describe the tempo, both of which can be used to decide what colour to use in the

visualisation as pitch and tempo, or fastness, are terms Kandinsky has used in his colour theories.

It's slightly harder to calculate the tone of an audio, but we're using the spectral complexity mean to do this as it takes into consideration the amount of peaks in the audio spectrum and could thus be used to indicate the tone which is defined by its attacks and decays. The tone can also be used to decide a colour to be used, as it can describe the mood of an audio.

There are also features which can be used to describe the form of the visualisation. These are for example the danceability and dynamic complexity, which estimate the complexity of the audio. More complex the audio, more complex the form.

## Conclusions

Features to be used:

- **Pitch** to estimate disturbance or movement from rest to intensive
- **Silence rate** to estimate the amount of silence
- **BPM** (Beats per minute) to estimate tempo from fast to slow, and mood from restless to calm
- **Danceability rate** to estimate mood from happy to sad
- **Tonal Chords change rate** to estimate complexity of the instrument.
- **Dynamic complexity** to estimate form complexity

These features are calculated for each separate source, which will then give 1-5 different definitions for a specific colour and shape combinations. The definitions for these selections are explained in more detail in the visualisation process.

We've used the separated source files from both Spleeter and Demucs models and calculated audio features with Essentia MusicExtractor and compared their performance as

seen in Figures 9 - 11.

Audio 1					
Filename	vocals.wav	bass.wav	drums.wav	piano.wav	other.wav
Source separation model	Spleeter	Spleeter	Spleeter	Spleeter	Spleeter
Silence rate (60dB mean)	0,85772	0,34255	0,62404	0,62436	0,36003
Pitch salience mean	0,50495	0,48362	0,52095	0,43254	0,54145
BPM	113,90752	120,07733	90,51786	178,20625	178,20607
Tonal Chords changes rate	0,16127	0,11566	0,10272	0,08978	0,07667
Danceability	0,83855	1,18455	1,38626	1,02185	1,08775
Dynamic complexity	43,71941	15,38261	11,93147	23,03756	6,06026
Source separation model	Demucs	Demucs	Demucs	Demucs	Demucs
Silence rate (60dB mean)	0,92478	0,40209	0,54801		0,34029
Pitch salience mean	0,64680	0,56669	0,53282		0,55958
BPM	89,70044	120,33481	90,37322		178,20586
Tonal Chords changes rate	0,09059	0,11339	0,11436		0,08201
Danceability	0,83339	1,21405	1,54373		1,07608
Dynamic complexity	30,29458	15,82897	10,97065		6,67887

Figure 9. Audio 1 comparison of source files when computing audio features with MusicExtractor

Audio 2					
Filename	vocals.wav	bass.wav	drums.wav	piano.wav	other.wav
Source separation model	Spleeter	Spleeter	Spleeter	Spleeter	Spleeter
Silence rate (60dB mean)	-	-	-	0,90887	0,61903
Pitch salience mean	-	-	-	0,67383	0,50447
BPM	-	-	-	151,72565	143,24777
Tonal Chords changes rate	-	-	-	0,09622	0,03822
Danceability	-	-	-	0,89371	0,93059
Dynamic complexity	-	-	-	63,59962	9,89920
Source separation model	Demucs	Demucs	Demucs	Demucs	Demucs
Silence rate (60dB mean)	-	-	-		0,61947
Pitch salience mean	-	-	-		0,50309
BPM	-	-	-		143,26654
Tonal Chords changes rate	-	-	-		0,03911
Danceability	-	-	-		0,93191
Dynamic complexity	-	-	-		9,74578

Figure 10. Audio 2 comparison of source files when computing audio features with MusicExtractor

Audio 3					
Filename	vocals.wav	bass.wav	drums.wav	piano.wav	other.wav
Source separation model	Spleeter	Spleeter	Spleeter	Spleeter	Spleeter
Silence rate (60dB mean)	0,91641	0,10684	0,41271	-	0,45218
Pitch salience mean	0,45401	0,32248	0,55740	-	0,48307
BPM	99,27650	163,83711	163,95576	-	164,01262
Tonal Chords changes rate	0,11063	0,11063	0,15355	-	0,05818
Danceability	0,97972	1,11610	1,34862	-	1,09674
Dynamic complexity	57,95213	6,97640	7,25262	-	7,64625
Source separation model	Demucs	Demucs	Demucs	Demucs	Demucs
Silence rate (60dB mean)	-	0,10529	0,50000		0,56439
Pitch salience mean	-	0,37006	0,57583		0,55266
BPM	-	163,49825	164,01822		164,02405
Tonal Chords changes rate	-	0,06271	0,12351		0,06319
Danceability	-	1,10438	1,34742		1,13270
Dynamic complexity	-	6,70669	7,76198		10,60107

Figure 11. Audio 3 comparison of source files when computing audio features with MusicExtractor

We're using the MusicExtractor features from individual sources to provide information

to calculate colour and shape definitions when combined with Kandinsky's colour theories.

## 6 AUDIO VISUALISATION LIBRARIES

To be able to understand what is currently and easily possible, we've investigated audio data visualisation possibilities with some of the most common python libraries.

### 6.1 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualisations in Python (Matplotlib 2022). With it, it's easy to visualise sound spectrum, frequency, and intensity as amplitude by using the spectrogram and waveform visualisations.

Waveform is a basic way for giving a graphical representation of a sound wave. The waveform depicts sound amplitude which moves (through a medium) over time. We can visualise this with Matplotlib waveform function, as seen in Figure 12.

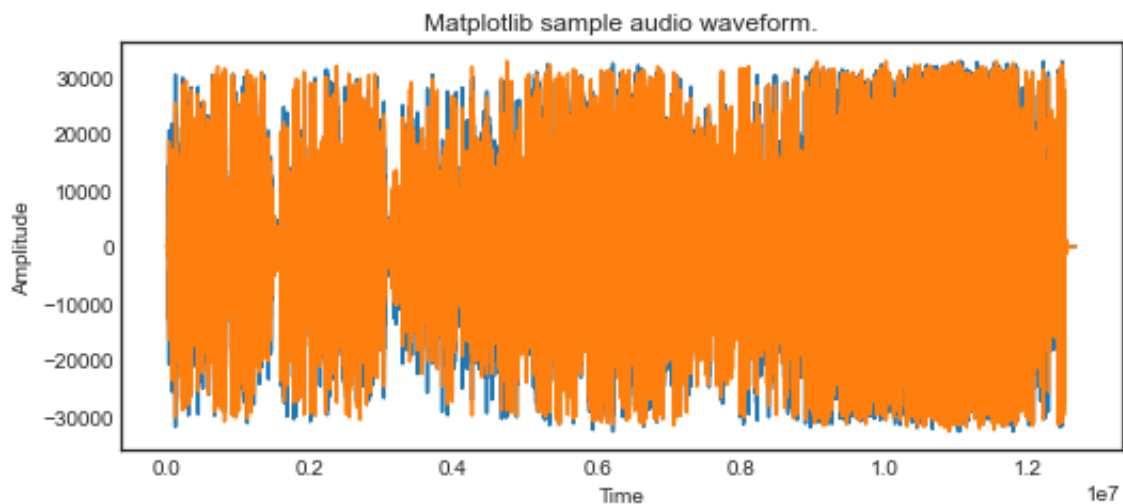


Figure 12. Matplotlib sample audio waveform.

A spectrogram visualises the sound spectrum in terms of frequency and amplitude over time. Frequency is measured in hertz (Hz), usually labelled on the left side, and amplitude (dB) is represented as the colour intensity, usually labelled on the right side of the spectrogram graph.

Based on the Matplotlib spectrogram visualisation, as seen in Figure 13, we can view how the low frequency sounds are of higher amplitude throughout the audio, as the red is more intense in the lower region of the graph.

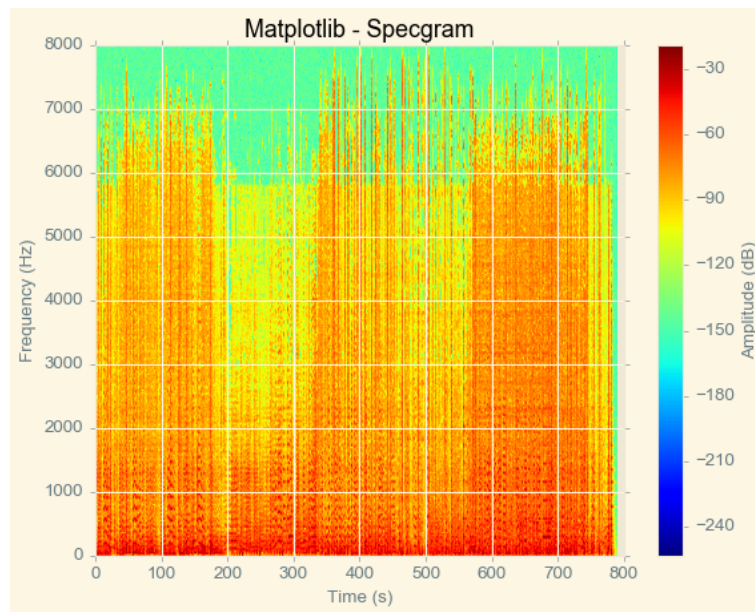


Figure 13. Matplotlib spectrogram visualisation of an audio.

## 6.2 Seaborn

Seaborn is a Python data visualization library based on Matplotlib and integrates closely with pandas data. It provides a high-level interface for drawing attractive and informative statistical graphics. (Seaborn 2022a)

Seaborn provides different styles for a Matplotlib waveform, but as a default, they are not very versatile. However, with Seaborn it's easy to switch between different visual representations by using a consistent dataset-oriented API. (Seaborn 2022b)

Seaborn has an easy functions for replotting the same data using a different graphical plot visualisation which can be seen in Figures 14 - 15. This might be very useful when trying to visualise same data with emphasis on different variables.

## 6.3 Librosa

Librosa can be used to visualise some audio sound elements not available in Matplotlib, as it's a python package specifically built for music and audio analysis. Librosa has many elements to be used when creating music information retrieval systems. It's to be noted that all of Librosa's plotting functions rely on Matplotlib. (Librosa 2021c)





Figure 14. Seaborn replotting function used to plot data.

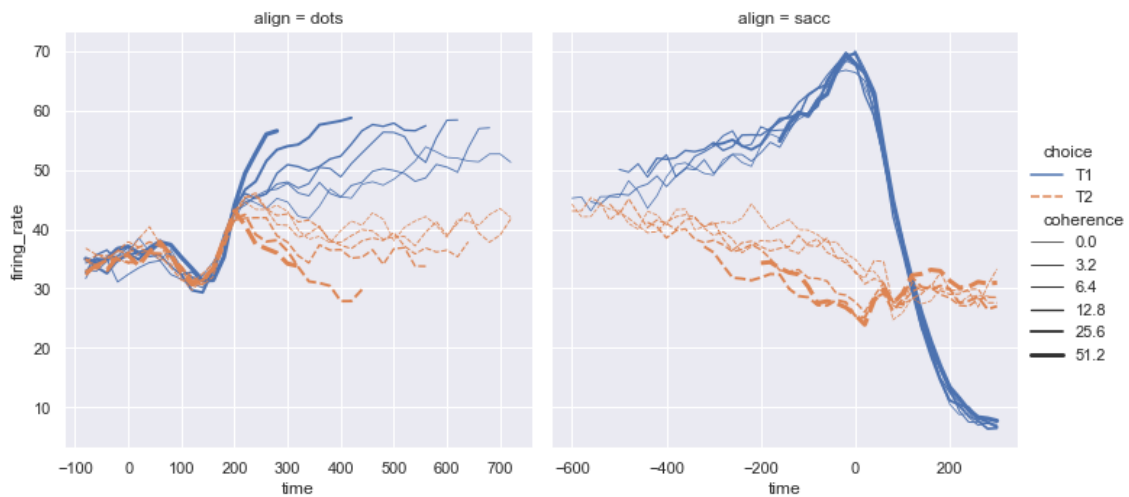


Figure 15. Seaborn replotting function used to replot the data seen in Figure 14 into different graphical plots

Compared to the default Matplotlib waveform of an audio, the Librosa waveshow (Figure 16) gives possibility to differentiate monophonic, stereo and harmonic components of the audio.

The frequency of an audio can also be visualised with Librosa using a Short-Time Fourier Transform (STFT) plot (Figure 17), which represents a signal in the time-frequency domain by computing discrete Fourier transforms (DFT) over short overlapping windows. (Librosa 2021b)

Librosa's specshow plot can be used to create various visualizations of spectro-temporal

data, as seen in Figure 18, and Librosa also has ready methods of visualising audio pitch. We can use the `piptrack` to give an audio pitch tracking on thresholded parabolically-interpolated STFT (Librosa 2021a).

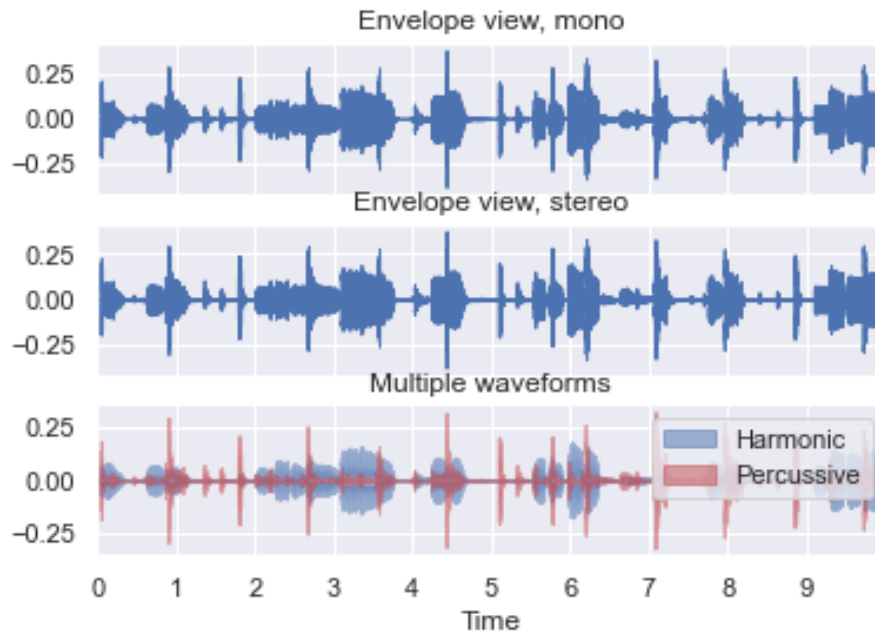


Figure 16. Librosa's waveplot graph comparing waveforms in mono, stereo, harmonic and percussive signals.

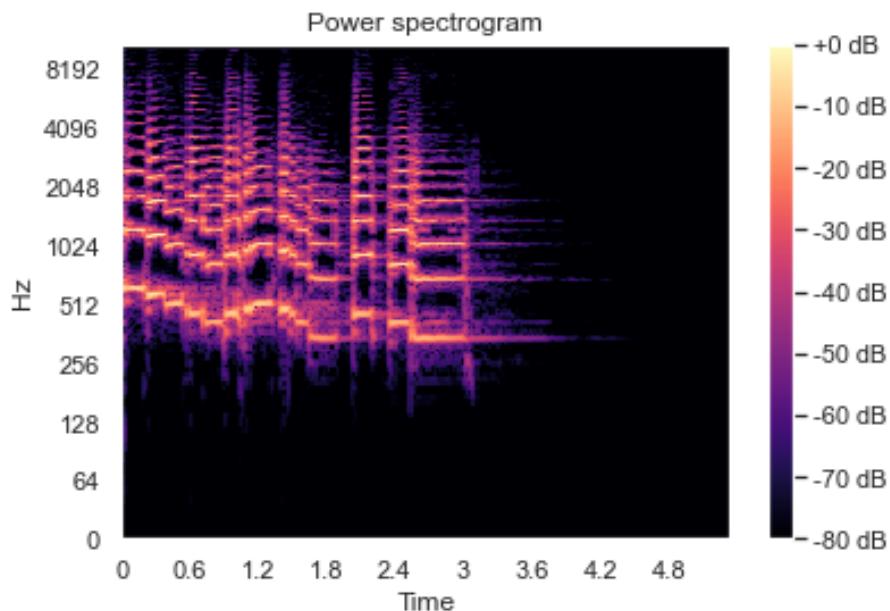


Figure 17. Librosa's Short-Time Fourier Transform graph

Visualising the same audio spectrograms with both Matplotlib and Librosa helps to see there are some core issues when interpreting the data, which relates to the default param-

eters of the plots in each library.

- The Matplotlib plot shows y-axis grid lines as a default on 1000 mark intervals, whereas Librosa's appears to show them based on the amount of information available.
- Librosa gives more insight for low frequency audio when it focuses the graph into those spectrums that have more data. But it doesn't make the graph any more readable.
- The colour values vary greatly, as Librosa only shows the values which are currently visible in the data. Matplotlib shows values under -120 dB, even though there doesn't seem to be such data when visually inspected.

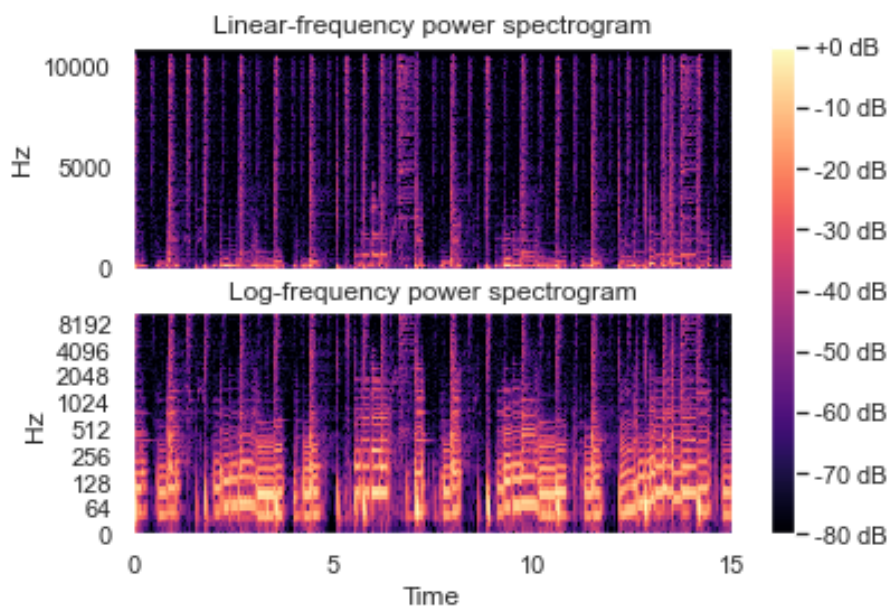


Figure 18. Librosa's specshow graph

## 6.4 Bokeh

Bokeh is a Python library for creating interactive visualizations. It helps to build beautiful graphics, ranging from simple plots to complex dashboards with streaming datasets. With a wide array of widgets, plot tools, and UI events that can trigger real Python callbacks, the Bokeh server is the bridge that connects to NumPy, Scipy, Pandas, Dask, Scikit-Learn, OpenCV, and more, to enrich interactive visualizations. (Bokeh 2022)

The Bokeh ridgeplot (Figure 19) could be used, for example, to visualise instrumental specific amplitudes, instead of the basic spectrogram. Bokeh hexbin (Figure 20) or hextile plots (Figure 21) could be used to visualise the estimation confidence by one or many instruments.

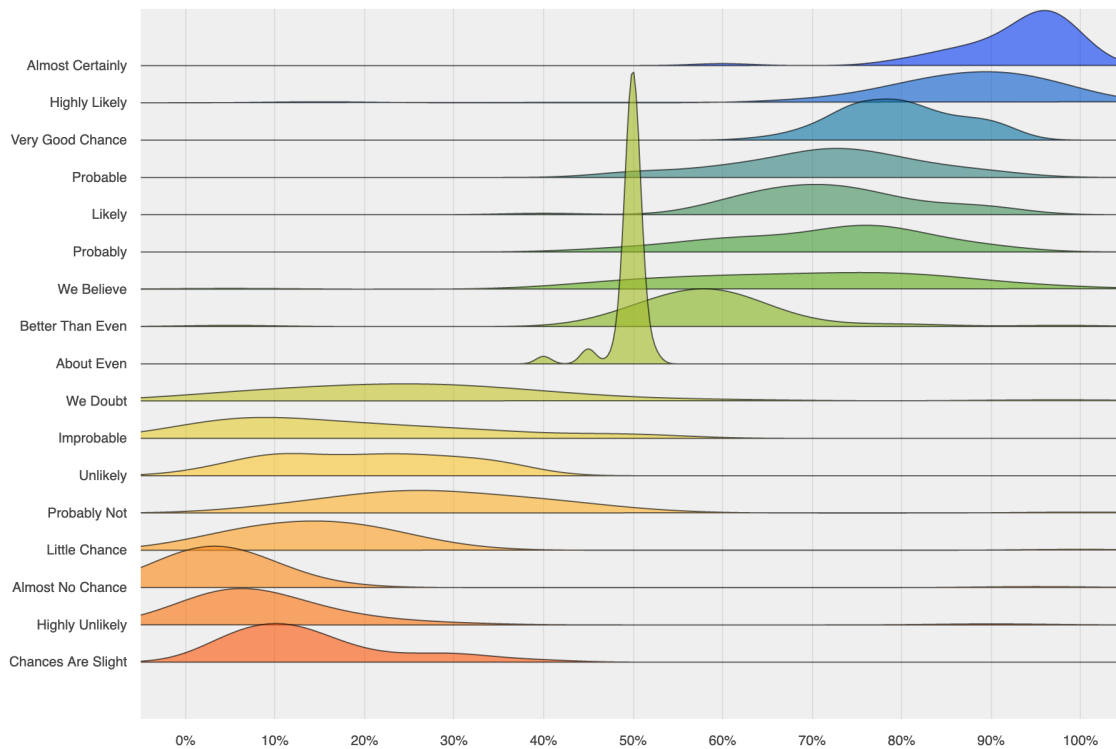


Figure 19. Example of Bokeh ridgeplot

Bokeh also offers versatile colour and shape graphs to visualise forms in RGB colour scale. The image\_rgba plot (Figure 22) could be great to visualise mood with colour background, as the gradual fade into separate colours supports having two different moods, we might get from the MSD and MTT datasets.

Bokeh plots have an option for plotting different shapes in specified RGB colour scales, which is an excellent way of visualising forms for a Kandinsky-themed art, as they are more abstract. For example, the Bokeh patch plot (Figure 23) can be used to visualise multiple forms in different colours at the same time.

Bokeh plot shapes can be built by using various visual shapes, or glyphs, which come in the forms of circles, ellipses, rectangles, wedges, arcs, béziers, lines, ovals, patches, and many other forms that are defined with x- and y-coordinate arrays.

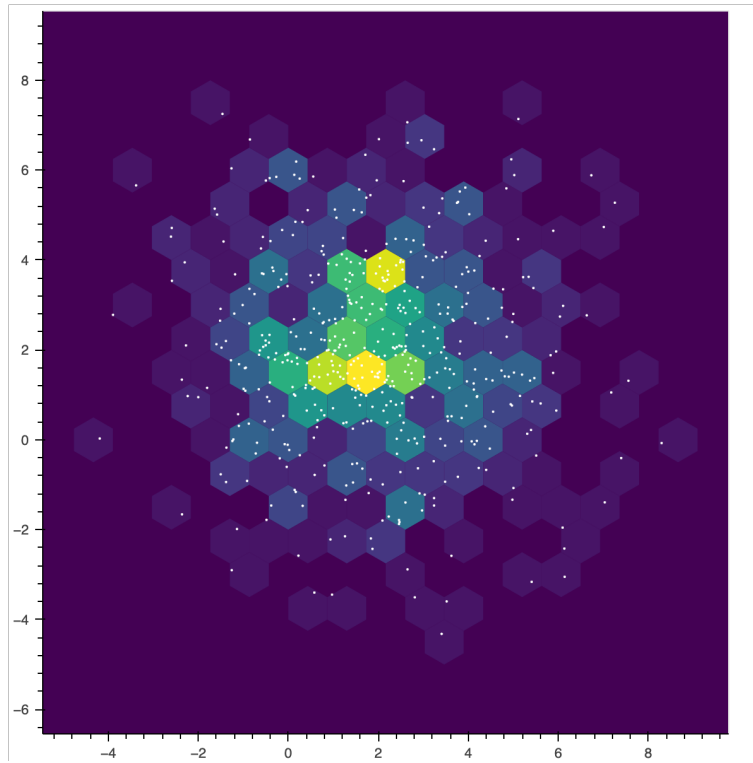


Figure 20. Example of Bokeh hexbin plot

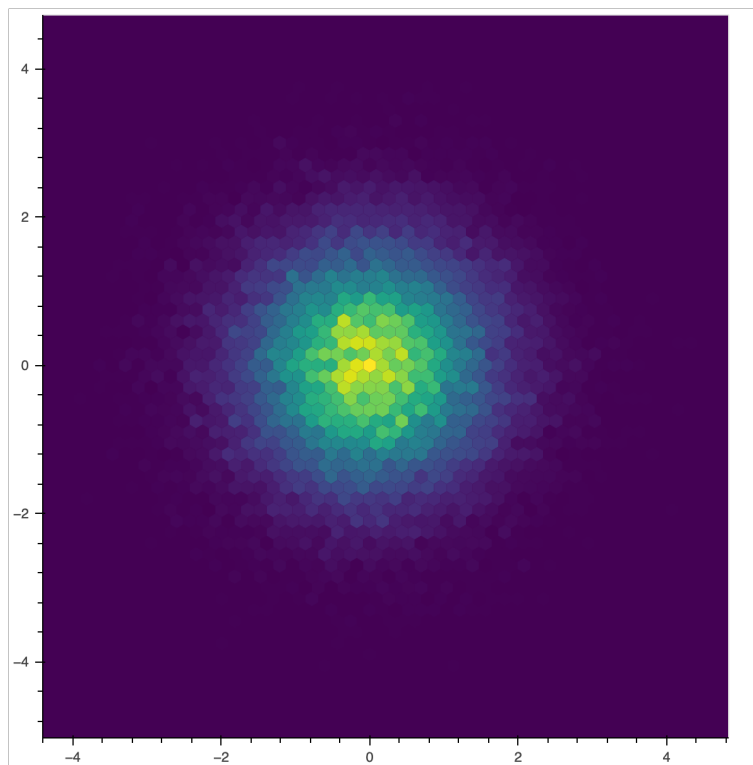


Figure 21. Example of Bokeh hextile plot

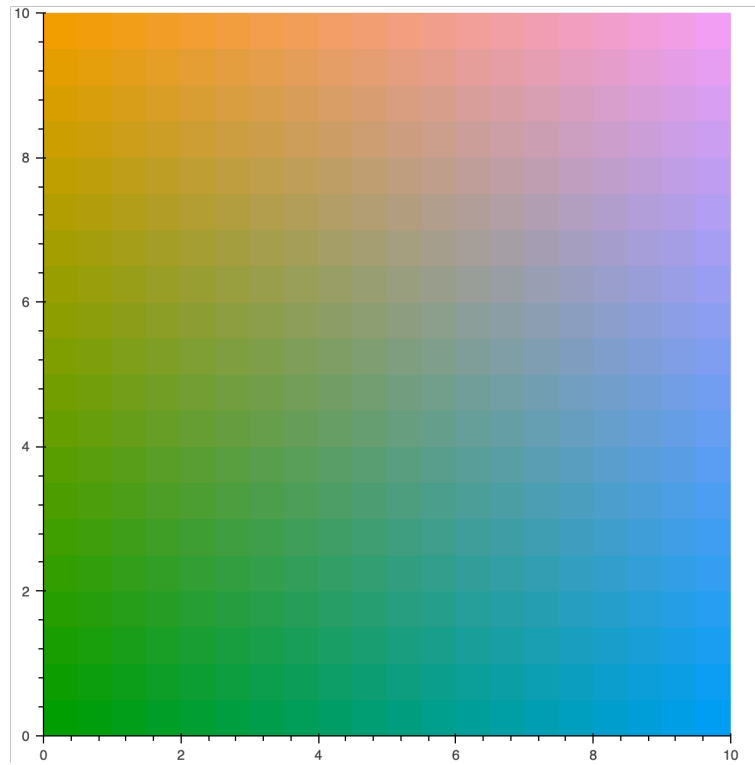


Figure 22. Example of Bokeh RGBA plot

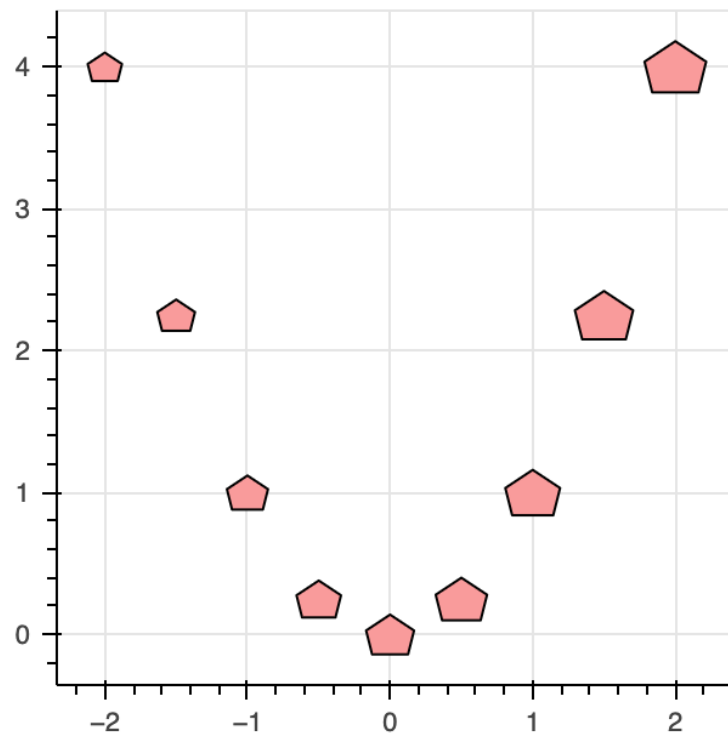


Figure 23. Bokeh patch plot

## 6.5 Conclusions

From all the afore mentioned python libraries, Bokeh seems to offer the most versatility and flexibility for this research. It has clear advances in having the possibility to visualise abstract forms very similar to Kandinsky's paintings, and it's also offering easier setup as the same colour coordinate arrays stored can be used in many Bokeh plots.

For these reasons, we've chosen to use Bokeh plots in the visualisation, especially the Bokeh rgba plot and the patch plot, which can be used to visualise the overall mood with background colour and the shape of specific instruments with different glyphs.

## 7 KANDINSKY'S THEORIES FOR COLOUR AND FORM

The end product of this research aims to be a visualisation which would help the viewer to understand what Kandinsky could have imagined while he was painting his masterpiece *Yellow-Red-Blue*, as seen in Figure 24. The final visualisation will be an independent piece of art, but it will also be a part of a simulation of what the process of creating *Yellow-Red-Blue* might've been like for him.

This journey, which started from a piece of music, continues now to define colours and shapes and aims to end in a combination of understanding music through the viewer's eyes.

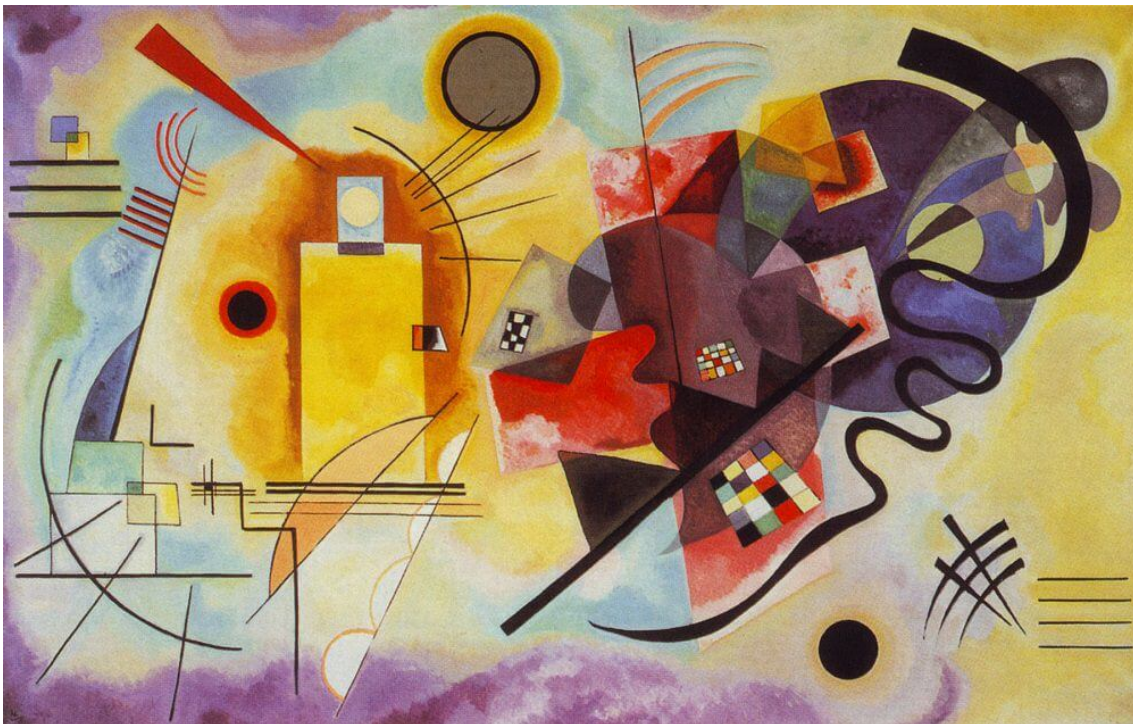


Figure 24. *Yellow-Red-Blue* by Wassily Kandinsky (Wassily-Kandinsky 2022)

It's fascinating how different perceptions people can have of the same colour. For instance, a quick glance to a tree might tell you that the tree's bark is brown, but upon a closer inspection it's actually a mixture between reds, blues, and greens. We can also never truly know in what shade of colour another human sees the same tree, as there are so many things affecting our observations.

This is why, before heading to define colours for the final visualisation, we need to take



a look at the foundation of Kandinsky's colour theories, and why he felt that colour and shape are spiritually linked to the human soul.

As Kandinsky wrote his book *Concerning the Spiritual in Art and Painting in Particular*, he explained his theory how art is just an abstraction of thought, which could be lead to purely artistic composition. (Kandinsky 2020) The way to get there is by understanding the languages of form and colour, which are the weapons of painting. He wanted to find a way to vibrate the human soul by his use of form and colour. The aim was to get to true harmony, which he said would exercise a direct impression on the soul.

He described his guiding principles to be colour harmony, form harmony and the choice of object (abstract or organic). These three principles direct the final visualisation when deciding its composition:

- First, we'll use machine learning models to depict the mood of the music and use Kandinsky's colour definitions to match the mood.
- Second, we'll use machine learning models to separate the instrumental sources from the overall audio and use Kandinsky's form definitions to match the instrumental information.
- And third, we'll use the abstract form, as it's the chosen method of Kandinsky himself.

This research takes refuge in Kandinsky's mention that even though different combinations of colour and shape have different spiritual values, and many colours can be hampered by the form given to them, we might still, with some manipulation of colour and form, find fresh possibilities of harmony. (Kandinsky 2020)

## 7.1 Colour harmony

Kandinsky gives some colour definitions in his book *Concerning the Spiritual in Art and Painting in Particular*, which we've used as the basis for colour and shape definitions in the final visualisation.

The first things to pay attention to in deciding what colours to use are the relationships between warm and cold and light and dark. These define the appeal of all colours. Kandinsky's antithesis of primitive colours state three colour pairs that are linked to each other and two which define the relation of the colour used to light and dark. These colour pairs are blue and yellow, red and green, orange and violet. Light and dark are represented with white and black.

### **Blue and yellow**

Based on Kandinsky's theories, the relationship between warm and cold refers to an inclination to yellow or to blue. This affects the experience of approach or retreat in the viewer, or the movement to a direction.

Yellow is a warm and happy colour which invites the viewer to approach easily. However, it has insistent and even disturbing influence, which might reach the level of being aggressive. Yellow can represent some forms of mental illnesses, and has a deep link to humanity. In a musical instruments, yellow can represent happy notes of a violin or a shrill sound of a trumpet.

Kandinsky thought blue to be a heavenly colour which creates a feeling of rest. It also has a cold component that takes the viewer further away from humanity. The lighter the blue, the more distant and weak the viewer feels. The darker the blue, the closer to death it takes the viewer.

The relationship of blue towards light or dark can represent musical instruments such as flute, cello, thunderous double bass or an organ.

Yellow can be used to describe the pitch of a sound, as the intensity of it increases the shrillness of the musical note. Blue can be used to describe the tone of a sound. We will have an estimation for the pitch and tone from the MusicExtractor features.

### **Red and green**

Kandinsky thought red as warm, determined, powerful and mature. It has a strong independent glow, which makes it very attractive. He also said that red by itself can't ever

be sad, which makes its presence or absence a good way to depict happy or unhappy sounds.

The relationship of red towards light or dark, and warm or cold, can represent musical sounds such as middle tones of a cello, singing notes of a violin and strong triumphant ringing of trumpets.

As green is formed from the intersection of blue and yellow, where movement ceases, Kandinsky felt it to be the most restful of the colours. Green can be viewed as motionless, passive, even wearisome, self-satisfied, immovable and narrow.

The relationship of green towards light or dark will indicate either the feeling of equanimity, or calmness in the middle of the storm, or restfulness, freedom of any disturbances. The difference comes not from the calming green itself, but from its counterparts. Green can represent musical sounds, such as the placid middle notes of a violin.

Red could be used to describe danceability, and green the tempo of a sound. We'll have an estimation for the danceability rate and BPM/tempo from the MusicExtractor features.

### **Orange and violet**

Orange is the combination of red and yellow. For Kandinsky, the characteristics of orange are those of red, but brought closer to humanity. To feel orange is to be a very self-confident man, and it can represent musical sounds, such as the notes of an old violin.

Violet is the combination of red and blue. Its characteristics are those of red, but withdrawn from humanity. In Kandinsky's theories violet is connected to sadness, ailing, old age and mourning. Violet can represent the English horn or deep notes of wood instruments.

The relationship between orange and violet could be used to describe the dynamic complexity of a sound. We'll have an estimation for the dynamic complexity from the MusicExtractor features.

## **White and black**

Finally, we come to the definitions of light and dark, or white and black as they are represented in Kandinsky's colour theories. The shade of the colour toward light (white) or dark (black) will emphasize the movement of the colour that is given light or darkness to, by being either dilating (light) or contracting (dark).

Kandinsky felt white to represent a harmony of silence, a temporal pause in the music that is full of expectations. One should be careful when using white as it mutes other colours.

Black, however, is the profound and final pause in the music for Kandinsky, where no more possibilities exist. Black silence is burnt out, motionless as death. Using black will make other colours stand out, it exaggerates them.

The combination of these two will result in grey, which is also silent and motionless, with no potential for activity.

We can have an estimation for the silence rate from the MusicExtractor features.

## **7.2 Form harmony**

For Kandinsky, the colour of an object in a canvas defined the inner need of the human soul, and the form he chose for this object was the outward expression of the inner need. This meant that the shape he gave to the object was dependent on the purpose of the colour. The form was chosen to amplify the inner need, in correlation to other objects on the canvas.

He could for example connect keen colours to sharp forms, such as yellow triangles, to create a feeling of passionate joy. And soft, deep colours to round forms, such as blue circles, to describe peaceful movement.

We can use estimations for sound tempo and dynamic complexity from the MusicExtractor, to select forms of individual objects.

### **7.3 Choice of object**

Kandinsky outlined that the choice of object refers to either abstract or organic object, where abstract would be for example a circle or a triangle and an organic object represent humans or animals.

We've chosen the object to be abstract as it follows Kandinsky's path, and for him "the more abstract is form, the more clear and direct is its appeal" (Kandinsky 2020). This will guide our research on how to combine the colours and shapes I've chosen for each mood and/or instrument.

### **7.4 Composition**

Even though, the three before-mentioned principles affect the whole composition, there are still a few guidelines to follow. It's important to understand that while each component of the final visualisations will have its own colour and shape, when arranging the components to form a one whole picture, the singular components will affect each other's harmony.

This means, there's still the question of the composition to solve. Kandinsky refers to this to "creation of the various forms which, by standing in different relationships to each other, decide the composition of the whole" (Kandinsky 2020).

Before adding the different components together, we must decide a way to optimise their ideal harmony, which might alter according to the relationship of the components. Even a slight approach or withdrawal from one component to another may affect the ideal harmony in a positive or a negative way.

We can use the mood labels from the VGG model to create a background visualisation for the whole composition, and then add the individual forms on top of this. We'll complete the final composition and combination of visualisations in a separate image processing software.

## 8 FINAL COMPOSITION

The audio used to create the final output is a song called *My Type* from the band Saint Motel, released in 2014. The music video for the song can be seen on YouTube <https://youtu.be/IyVPyKrx0Xo> (Saint Motel 2022).

The song is labelled into progressive pop, alternative rock, electronic, indie rock, pop and rock genres (MusicBrainz 2022). The duration of the audio is 3:25. The waveform for the song can be seen in the Figure 25, which shows that the song has quite consistent amplitude during its play.

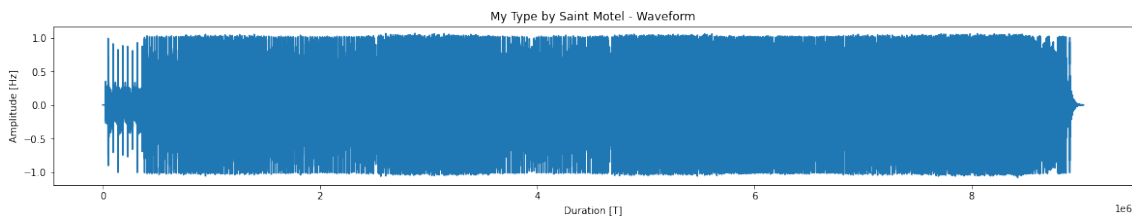


Figure 25. Waveform visualisation of *My Type* by Saint motel

### 8.1 Background

First, we use VGG model to separate mood depictees for the audio from both MSD and MTT datasets. The results can be seen in Figures 26 and 27. We'll use these to calculate colours for the background image of the whole visualisation.

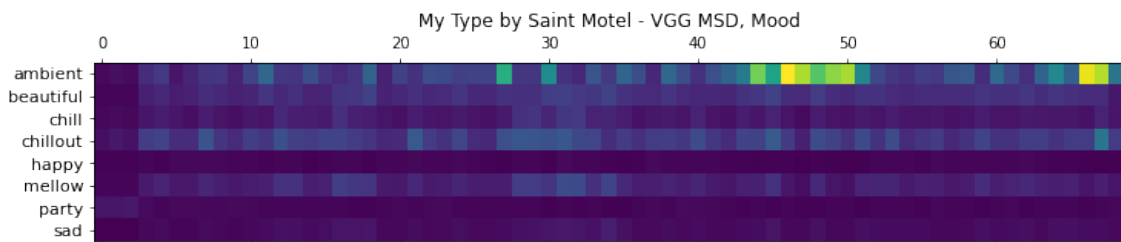


Figure 26. Top VGG MSD mood labels

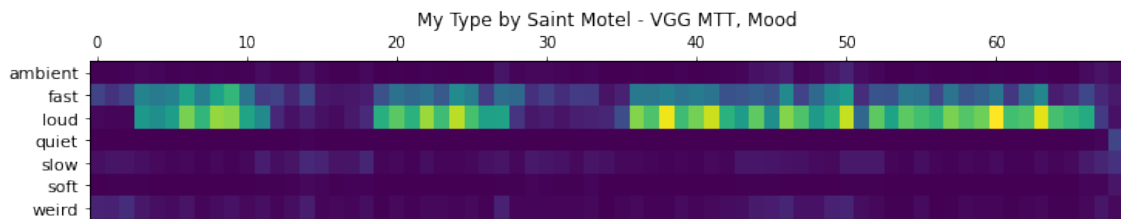


Figure 27. Top VGG MTT mood labels

Using Kandinsky's colour theories, we're giving estimated settings for the background colours. They are defined in the rgba scale or in the relationship between red, green, blue, and alpha. we've made a comparison between the moods gained from VGG model and their respective colours according to Kandinsky.

- RED moods: fast, weird, happy
- GREEN moods: slow, chillout, chill
- VIOLET moods (not-green): quiet, sad
- BLUE moods: soft, mellow
- YELLOW moods (not-blue): loud, beautiful
- AMBIENT moods: mixture of all moods and colours

The VGG model used will give two estimations for colours; one from the MSD and the other from the MTT dataset. We're using both of these moods to create a gradient background of two or more colours, or in case of AMBIENT mood, all colours. The exception to this would be the case of having two moods which both represent the same colour.

The VGG model estimates that the most prominent moods to be "ambient" (MSD) and "loud" (MTT). These indicate that there should be all colours available ("ambient" mood) but emphasis on yellow ("loud" mood). Using Bokeh RGBA plot, the background for the whole visualisation (Figure 28.) is set to contain a combination of all colours, putting emphasis on yellow.

It's also good to note, that during the process for creating a background image, we've noticed that the amount of green is calculated differently than other colours. While using Bokeh RGBA plots, the amount of green can never be used as full 255 as others in the RGB colour scale are used. This is because full green in the RGB scale is closer to neon green, and is not coherent with Kandinsky's colour theories of green colour.

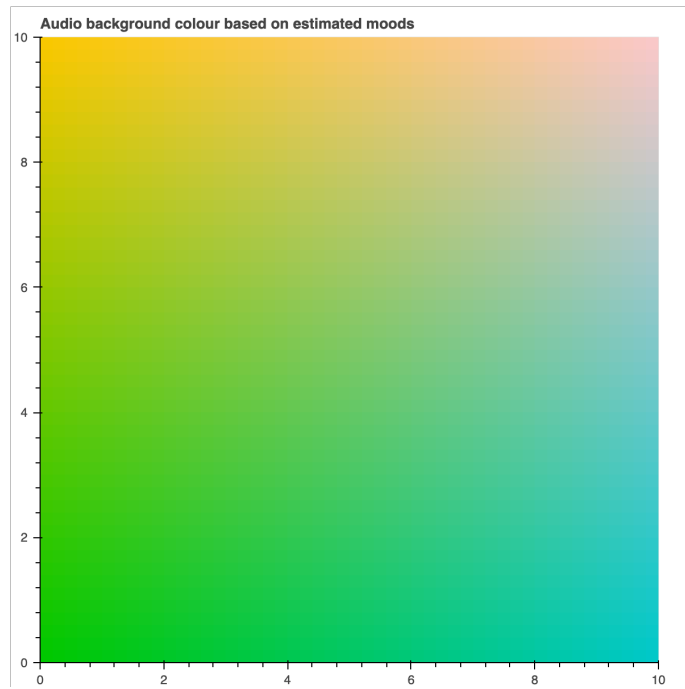


Figure 28. Background for the audio visualisation

## 8.2 Forms

Using Spleeter source separation model, we've separated the audio into five different instrumental sources: vocals, bass, drums, piano and other. The waveforms for these sources are represented in the Figure 29.

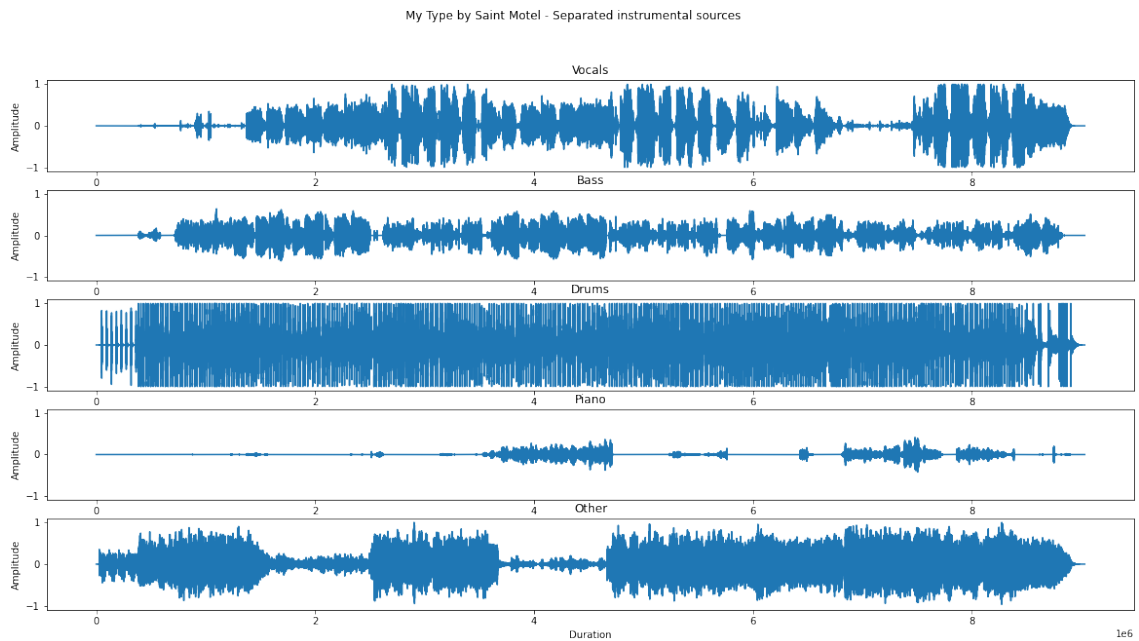


Figure 29. Separated instrumental sources for the audio.



From the waveforms, we can see that there is data available for all the sources. We can also verify that each waveform is unique and featuring a specific source only. We can also make rough estimates of the amplitude increase and decrease for specific instruments during the audio, and use this information when we are finalising the visual compositions.

Next, the audio and its separated sources are run through the MusicExtractor to extract audio features for silence rate (60dB mean), pitch salience mean, beats per Minute (BPM), tonal chords changes rate, danceability rate, dynamic complexity and standard deviance of dissonance (Table 7). These are used in the colour and shape definitions.

Table 7. Audio features extracted for the whole audio and separate sources with MusicExtractor.

Source	Silence rate	Pitch salience mean	BPM	Tonal Chords changes	Danceability	Dynamic complexity	Dissonance (stdv)
Full audio	0,0634	0,5330	117,9864	0,0406	1,1618	3,3312	0,0240
vocals.wav	0,4619	0,4802	117,9282	0,1104	0,9673	15,6308	0,0857
bass.wav	0,2656	0,3050	117,5507	0,0986	0,9681	12,0352	0,1356
drums.wav	0,3806	0,4705	118,0546	0,1022	1,6621	7,5051	0,0338
piano.wav	0,7276	0,5584	117,3827	0,0440	0,9247	42,4802	0,0941
other.wav	0,2656	0,5369	117,5352	0,0233	0,9573	6,1704	0,0565

With these definitions in mind, it's possible to calculate the forms for individual instrumental sources and give them unique colours, and to decide how to visualise forms for each instrumental source, we've taken a look at the dynamic complexity and dissonance for each source (Figure 30) and can see that "piano" source is clearly more complex than other sources, whereas "bass" is highlighted as having the most dissonance. However, the amount of dissonance for any source is relatively low.

From the dynamic complexity, we can see the average absolute deviation from the global loudness and from the dissonance the perceptual roughness of the audio spectral peaks. We've used these to define the shape of the audio on the basis that the bigger the relationship between these variables, the more angles the form shape has.

As the exact values are not important in the visualisation, we're aiming to find out the re-

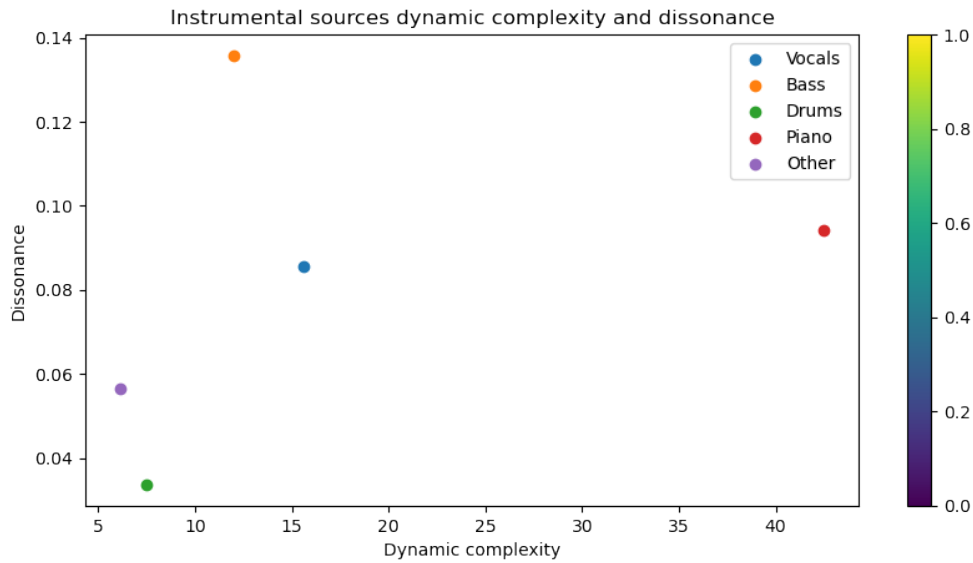






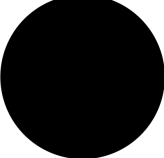
Figure 30. Instrumental sources dynamic complexity and dissonance.

relationship of the different forms, and attributes to use in their colours. For this, we've defined each source to have a variable "form" to give an estimation of its complexity.

$$form = abs(dynamic\_complexity/dissonance\_stdv)$$

The form type will vary between shapes Kandinsky used in his Yellow-Red-Blue painting (Figure 24) based on the "form" variable given: line or arc, circle, ellipse, wedge, triangle, rectangle, distorted rectangle or combinations of many forms. Having these estimations in place, it's possible to calculate the forms for the separated sources, as shown in Table 8.

Table 8. Forms for different instrumental sources.

Vocal	Bass	Drums	Piano	Other
				

### 8.3 Colours

After having the specific forms, they have to be given colours. They are set with the danceability rate, tonal chords changes rate, bpm, pitch salience (mean) and the silence rate (60 dB mean), gained from the source audio features, by scaling them to the RGB colour values from 0 to 255.

$$RED = source\_danceablity\_rate * ((255/danceability\_max) * 100)$$

$$GREEN = source\_bpm * ((150/bpm\_max) * 100)$$






$$BLUE = 255 - (source\_pitch * ((255/pitch\_max) * 100))$$

$$ALPHA = source\_silence * ((1/silence\_max) * 100)$$

Individual colour values for Red, Green and Blue are then either decreased or increased based on the moods gained from the VGG model, according to Kandinsky.

And finally, any values exceeding the RGB scale from 0 to 255 were capped to the maximum value of 255. Results for the colours are seen in the Table 9.

Table 9. Colours for different instrumental sources.

Vocal	Bass	Drums	Piano	Other
				
Mood: "slow"	Mood: "quiet"	Mood: "fast"	Mood: "quiet"	Mood: "loud"

### 8.4 Composition

Composition of the final visualisation (Figure 31) is done using the background colours and individual forms with colour values together in a digital painting application called Procreate. The goal has been to combine different elements together in a way that they will complement each other, and the forms will support the colours selected and colours

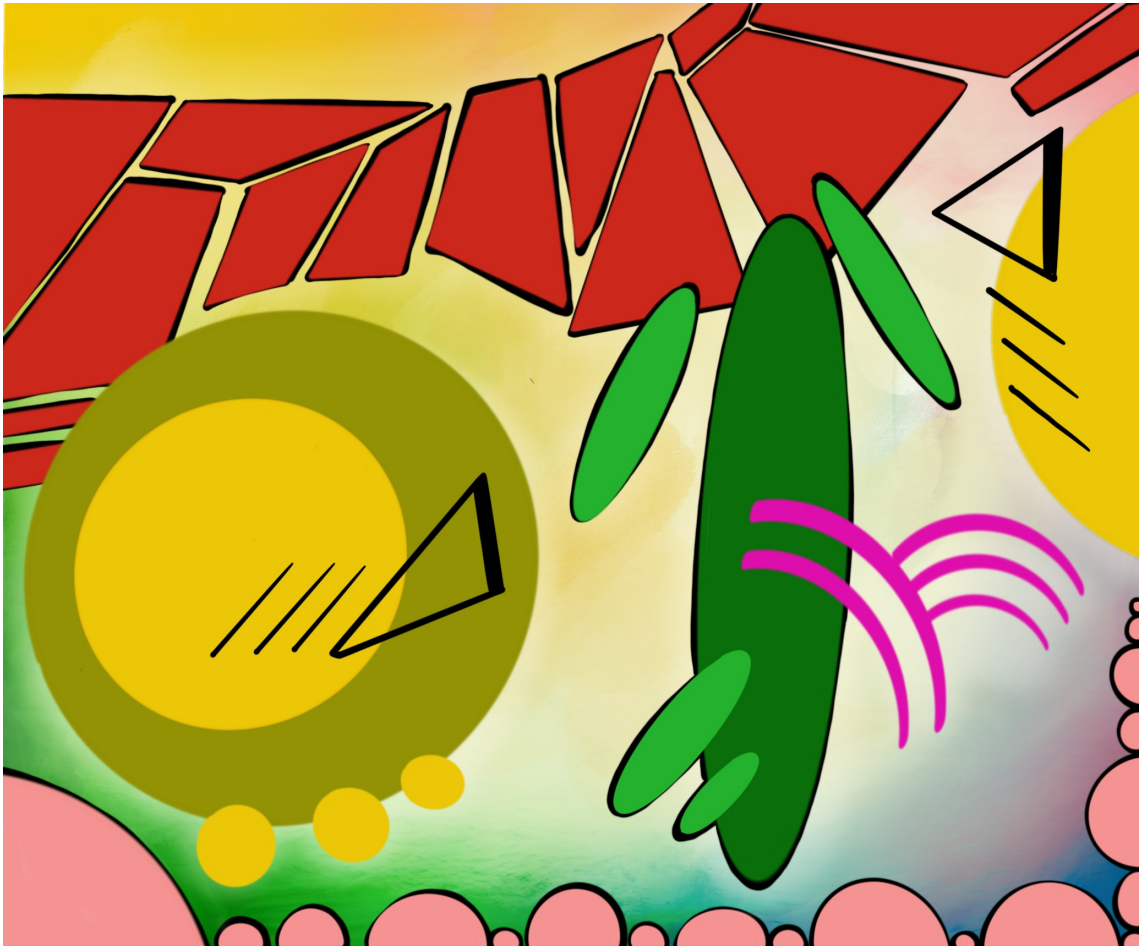
will support the forms.

The background is rich with different colours, with the emphasis on yellow, as the overall mood of the audio was ambient.

The green ellipses are representing the vocalisation which is calm but has ongoing movement. Bass is quietly heard, but it's consistently the backbone of the audio, and it's this prominently shown in pink arks, almost in the centre of the composition. Soft pink wedges on the edge of the composition illustrate how drums create the rhythm of the audio.

Piano doesn't come out too loud against its counterparts as it's very independent, but it's intricately composed and changes in variation throughout the audio. Its red quadrilaterals are constantly moving forward, even though they are changing their shape.

And finally, the other sources are represented in olive circle, which is hovering in the background. This is connected to sudden yellow, loud sounds of trumpets and other instruments, and black lines either suddenly adding silence to other forms or supporting their calculated pauses.



*Figure 31. Final visual composition of the song My Type*

## 9 CONCLUSIONS

The aim of this research was to understand how the process of hearing music and creating visual art from it, as experienced by Kandinsky, could be recreated with data and machine learning models. In this thesis we've estimated genres, moods and features for the audio, to combine them with Kandinsky's colour and form theories. Then we've used these findings to guide us in creating a whole visual composition from a musical audio.

It's to be noted, that this research focuses on a very small amount of data available for any musical audio. We've used 5 instrumental sources and the whole audio in our estimations. However, there are hundreds of other variables that could be used to recreate Kandinsky's visions. Also, the audible part of this data cannot directly be turned into variables, as it's heard by its audience and is thus affected with the possibility of human error.

The original scope of the project changed during this research process. One of the most early ideas was to use real-time audio data and create a custom machine learning model by training music focused data from Google AudioSet audio database. However, this idea had to be scrapped as the Google AudioSet data is stored in Tensorflow .tfrecords files and there were many difficulties with then just updated Tensorflow 2.0 framework.

In the beginning we also couldn't comprehend how much music and art theory would be needed to even understand the entire process, which caused the research to take a lot more time than expected.

We began from studying the basics of music and art theories, learning the basics of python programming and reviewing what possibilities were available in pre-trained machine learning models. As the knowledge of what was possible was relatively low at the time of starting this research, the original aim was soon to be seen too ambitious to be completed in the allocated time available.

Music information retrieval and source separation methods were completely unknown to us before starting this research. While there was an attempt to understand them by reading and participating in events such as the ISMIR conference organised by The International Society for Music Information Retrieval, only after tens of hours of practical testing, did

we start to have a glimpse of the process behind analysis of musical data.

In the beginning of this research, there was also hope to create real-time updating visualisations while the music was being played and streamed to the model. This was not achieved, partly because of the limitations of researchers skills at the time, and partly to the amount of data this type of processing would've inputted, which also set criteria to the hardware needed to properly process the amount of data in real-time.

What was created was an example of the process used to visualise musical data, utilising available machine learning models, audio features, and finding connections to those in the Kandinsky's art theories. This example contains also a final visualisation made using the ready colour and form definitions gained from the process, but it's composed by a human touch.

Even though the whole process of learning the basics, finding out the right paths after falling into many rabbit holes, and finally starting to understand the process and being able to streamline it to the point of outputting the definitions for abstract art, took what can be counted as over thousand hours, the final visualisation itself needed tens of hours to be completed.

There were several artistic restrictions while creating the final visual composition. Firstly, the final composition could not be completed by just copying Kandinsky's abstract style. As he explained, art is an abstraction of thought, and even though we might try to understand parts of the process experienced by Kandinsky, we cannot fully comprehend the volume of imagination which might go through inside an artist's mind while he is creating art.

We also have to consider the fact that he might have only been able to write down parts of his guiding principles, which we researched in his book *Concerning the Spiritual in Art and Painting in Particular*.

And of course, there is the element of the artist who is creating the final visual composition. How accurately they were able to or wished to use the readily given shape and form

definitions. What other elements they might've integrated into the visualisation, what is their personal artistic style, and how did they actually hear and experience the music itself.

The final visualisation is an independent piece of art and can't be dependent solely on Kandinsky's style, to be able to achieve its own artistic integrity. The vision, knowledge, and technique of the artist will affect the final visualisation as they're translating the data gathered into a composition of forms and colours.

Again, we do take respite in Kandinsky's words how it's possible to find fresh possibilities of harmony for each piece of art.

## **Future research questions**

Some research questions have arisen while doing this thesis. For example, the Spleeter and Demucs source separation models proved invaluable to this research, and it would be interesting to try to create a new model in the future, based on either of them, which could also take into consideration the need for visual output.

The colour and form calculations based on the separated instrumental sources can be further developed by taking into consideration the time locations where the instrumental amplitudes show significant changes. As the calculations are currently applied to features extracted from the whole instrumental audio, it also includes data which is filled with lower amplitudes, or full silence. This means that there's the possibility of definitions for colour and form to be more complex if these, less audible, parts of the data were removed. The other opportunity would be to segment the instrumental data to contain only parts which have consistent amplitudes or which form specific patterns.

There are also several practical improvements available for how the MusiCNN and VGG models were used in this research. Finding a dataset containing all necessary labels for the final visualisation could streamline this process tremendously. This could be achieved by training a new model, possibly with Google AudioSet data.

And of course, there's the original research question of how to turn this process into real-



time visualisation of musical data being streamed into the model while the music is being played live.

The final visualisation could be improved with many methods. For example, it could be tested how the final visualisation would be composed if an artist were to create it before and after learning of the data. As the composition was based on elements and colours received from an algorithm, creating the visualisation was forced into certain frame. Would the composition have changed, if the artist would've created it before learning of the insights found from the musical data.

The final composition could also be refined significantly by improving the integration between machine learning models and Kandinsky's theories, and if the artist creating it would have even close to Kandinsky's capabilities of understanding abstract art. Creating the final visualisation on canvas instead of digital format might bring up even more possibilities for improvement. This would require more research in art theory and creation, especially in its abstract aspects.

And finally, the process does raise the question of how Kandinsky himself would've painted the musical piece used in this research. We are left to wonder what it would've looked like from the master's brush.

## REFERENCES

- Bokeh. 2022, *Bokeh*. Accessed: 2022-01-09. Available: <https://docs.bokeh.org/>.
- Britannica. 2021, *Sound - Properties, Types, & Facts - Britannica*. Accessed: 2021-12-29. Available: <https://www.britannica.com/science/sound-physics>.
- Brixen, Eddy B. 2020, *Audio Metering: Measurements, Standards, and Practice*, Focal Press.
- Cooper, Matthew; Foote, Jonathan; Pampalk, Elias & Tzanetakis, George. 2006, Visualization in audio-based music information retrieval, *Computer Music Journal*, vol. 30, no. 2, pp. 42–62.
- Deezer. 2022, *Deezer Spleeter*. Accessed: 2022-01-29. Available: <https://github.com/deezer/spleeter>.
- Défossez, Alexandre. 2021, Hybrid Spectrogram and Waveform Source Separation, *arXiv preprint arXiv:2111.03600*.
- Denver Art Museum. 2021, *Wassily Kandinsky's Symphony of Colors*. Accessed: 2021-12-29. Available: <https://www.denverartmuseum.org/en/blog/wassily-kandinskys-symphony-colors>.
- Essentia. 2021, *Homepage - Essentia 2.1-beta6-dev documentation*. Accessed: 2021-12-29. Available: <https://essentia.upf.edu/>.
- Essentia. 2022a, *Algorithm reference - Dissonance (standard mode)*. Accessed: 2022-02-27. Available: [https://essentia.upf.edu/reference/std\\_Dissonance.html](https://essentia.upf.edu/reference/std_Dissonance.html).
- Essentia. 2022b, *Algorithm reference - DynamicComplexity (standard mode)*. Accessed: 2022-02-27. Available: [https://essentia.upf.edu/reference/std\\_DynamicComplexity.html](https://essentia.upf.edu/reference/std_DynamicComplexity.html).
- Essentia Labs. 2021, *A collection of TensorFlow models for Essentia*. Accessed: 2021-12-29. Available: <https://mtg.github.io/essentia-labs/news/tensorflow/2020/01/16/tensorflow-models-released/>.

- Google Arts & Culture. 2022, *Play a Kandinsky*. Accessed: 2022-02-26. Available: <https://artsandculture.google.com/experiment/play-a-kandinsky/sgF5ivv105ukhA?hl=en>.
- Ishibashi, Tatsuya; Nakao, Yuri & Sugano, Yusuke. 2020, Investigating audio data visualization for interactive sound recognition, In: *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pp. 67–77.
- Itoh, Kosuke; Sakata, Honami; Igarashi, Hironaka & Nakada, Tsutomu. 2019, Automaticity of pitch class-color synesthesia as revealed by a Stroop-like effect, *Consciousness and Cognition*, vol. 71, , pp. 86–91.
- Kandinsky, Wassily. 2020, *Concerning the Spiritual in Art and Painting in Particular*, Barakaldo Books.
- Khulusi, Richard; Kusnick, Jakob; Meinecke, Christofer; Gillmann, Christina; Focht, Josef & Jänicke, Stefan. 2020, A survey on visualizations for musical data, In: *Computer Graphics Forum*, vol. 39, Wiley Online Library, pp. 82–110.
- Librosa. 2021a, *Librosa.core.piptrack*. Accessed: 2021-12-29. Available: <https://librosa.org/doc/main/generated/librosa.piptrack.html>.
- Librosa. 2021b, *Librosa.core.stft*. Accessed: 2021-12-29. Available: <https://librosa.org/doc/main/generated/librosa.stft.html>.
- Librosa. 2021c, *Using display.specshow*. Accessed: 2021-12-29. Available: <https://librosa.org/doc/main/generated/librosa.display.specshow.html>.
- MA Rouf. 2019, *Feather*. Accessed: 2022-01-09. Available: <https://www.youtube.com/watch?v=4EX4yo0w0zA>.
- MacLeod, K. 2022, *Upbeat Forever*. Accessed: 2022-01-09. Upbeat Forever Kevin MacLeod (incompetech.com). Available: <https://www.chosic.com/download-audio/27246/>.
- Matplotlib. 2022, *Matplotlib*. Accessed: 2022-01-09. Available: <https://matplotlib.org/>.
- Merriam-Webster. 2021a, *Art Definition & Meaning*. Accessed: 2021-12-29. Available: <https://www.merriam-webster.com/dictionary/art>.

- Merriam-Webster. 2021b, *Data Definition & Meaning*. Accessed: 2021-12-29. Available: <https://www.merriam-webster.com/dictionary/data>.
- Merriam-Webster. 2021c, *Sound Definition & Meaning*. Accessed: 2021-12-29. Available: <https://www.merriam-webster.com/dictionary/sound>.
- Merriam-Webster. 2021d, *Timbre Definition & Meaning*. Accessed: 2022-01-22. Available: <https://www.merriam-webster.com/dictionary/timbre>.
- MusicBrainz. 2022, *Release group "My Type" by Saint Motel - Tags - MusicBrainz*. Accessed: 2022-01-30. Available: <https://musicbrainz.org/release-group/79dc899a-744f-4145-9386-af2260fafed8/tags>.
- Olson, Harry Ferdinand. 1967, *Music, physics and engineering*, vol. 1769, Courier Corporation.
- Saikkonen, Lauri et al.. 2020, Structural analysis of recorded music.
- Saint Motel. 2022, *My Type*. Accessed: 2022-01-30. Available: <https://youtu.be/IyVPyKrx0Xo>.
- Seaborn. 2022a, *Seaborn*. Accessed: 2022-01-09. Available: <https://seaborn.pydata.org/>.
- Seaborn. 2022b, *Seaborn*. Accessed: 2022-01-09. Available: <https://seaborn.pydata.org/introduction.html>.
- SigSep. 2021, *MUSDB18 | SigSep*. Accessed: 2021-12-29. Available: <https://sigsep.github.io/datasets/musdb.html#musdb18-compressed-stems>.
- Van Elferen, Isabella. 2020, *Timbre: paradox, materialism, vibrational aesthetics*, Bloomsbury Publishing USA.
- Vivaldi, A. 1723, *The Four Seasons*. Accessed: 2022-01-09. John Harrison, violin, with Robert Turizziani conducting the Wichita State University Chamber Players. Live, unedited performance at the Wiedemann Recital Hall, Wichita State University, 6 February 2000. Music by Antonio Vivaldi composed 1723 and published in 1725. Available: [https://freemusicarchive.org/music/John\\_Harrison\\_with\\_the\\_Wichita\\_State\\_University\\_Chamber\\_Players/The\\_Four\\_Seasons\\_Vivaldi](https://freemusicarchive.org/music/John_Harrison_with_the_Wichita_State_University_Chamber_Players/The_Four_Seasons_Vivaldi).

Wassily-Kandinsky. 2022, *Yellow-Red-Blue, 1925 by Wassily Kandinsky*. Accessed: 2022-02-28. Available: <https://www.wassily-kandinsky.org/Yellow-Red-Blue.jsp>.

Xu, Zhicun et al.. 2018, Audio Event Classification Using Deep Learning Methods.

# APPENDIXES

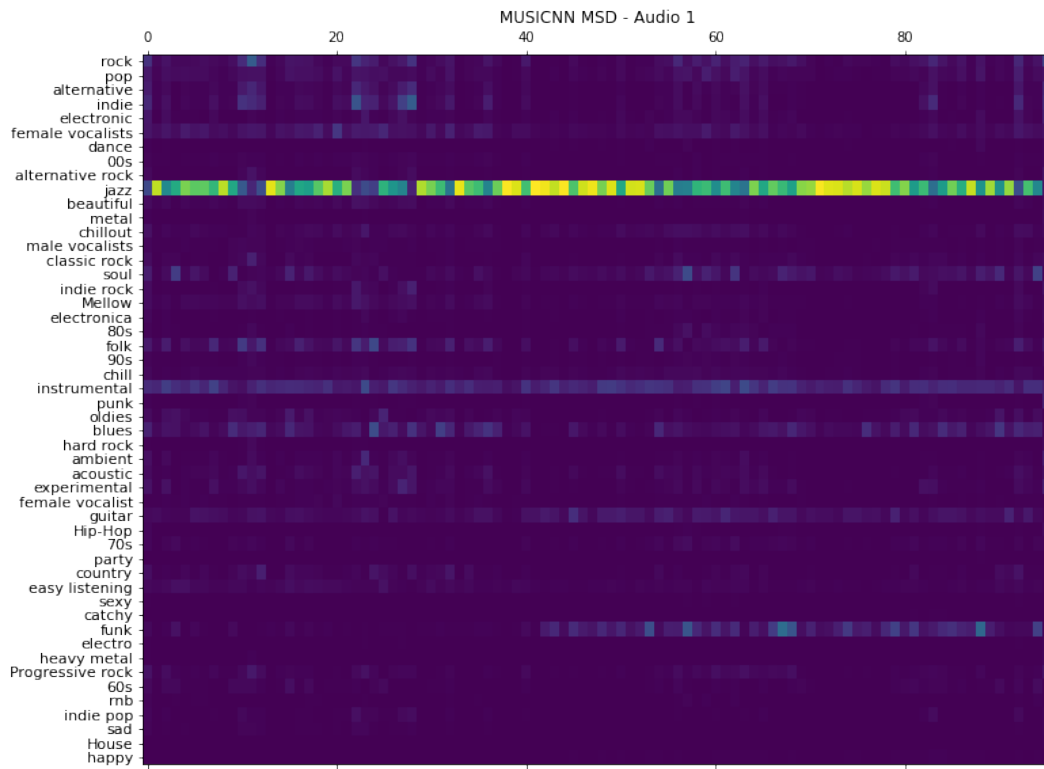


Figure 32. MusicNN MSD, audio 1

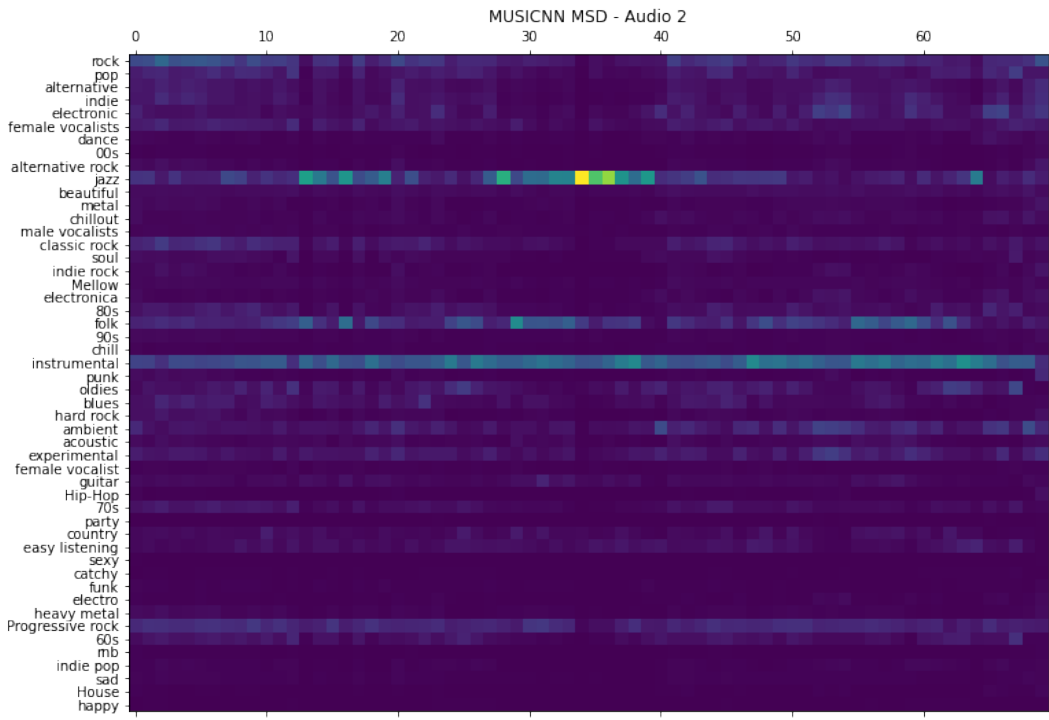


Figure 33. MusicNN MSD, audio 2

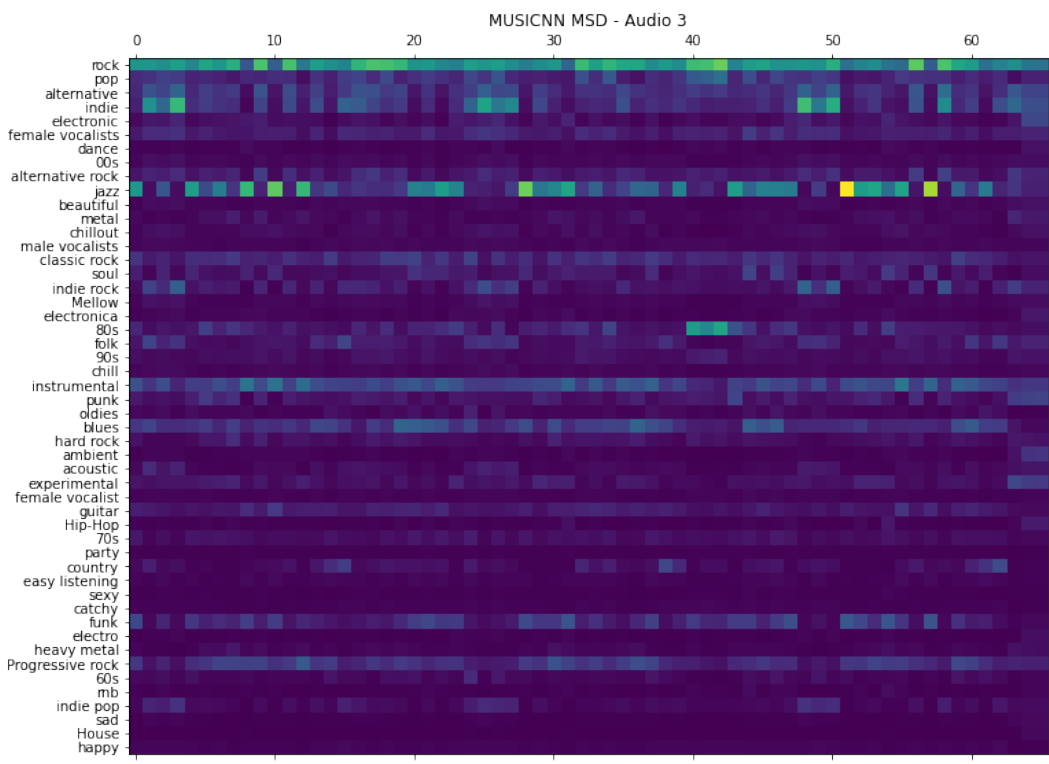


Figure 34. MusicNN MSD, audio 3

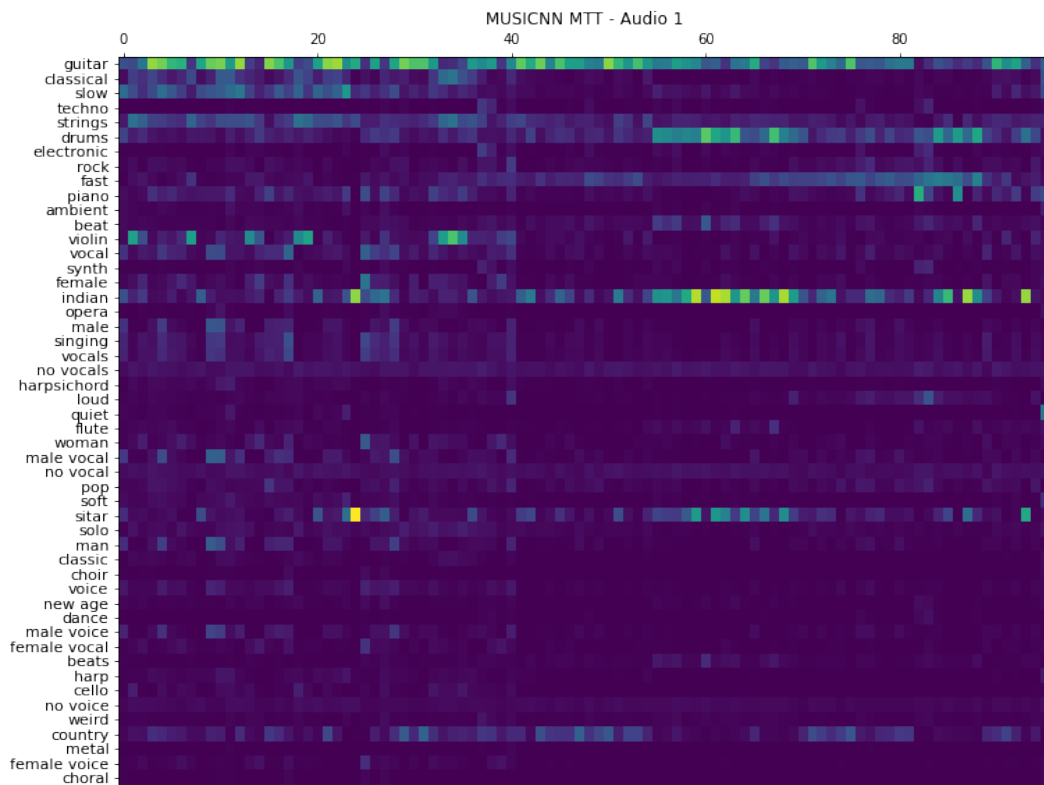


Figure 35. MusicNN MTT, audio 1

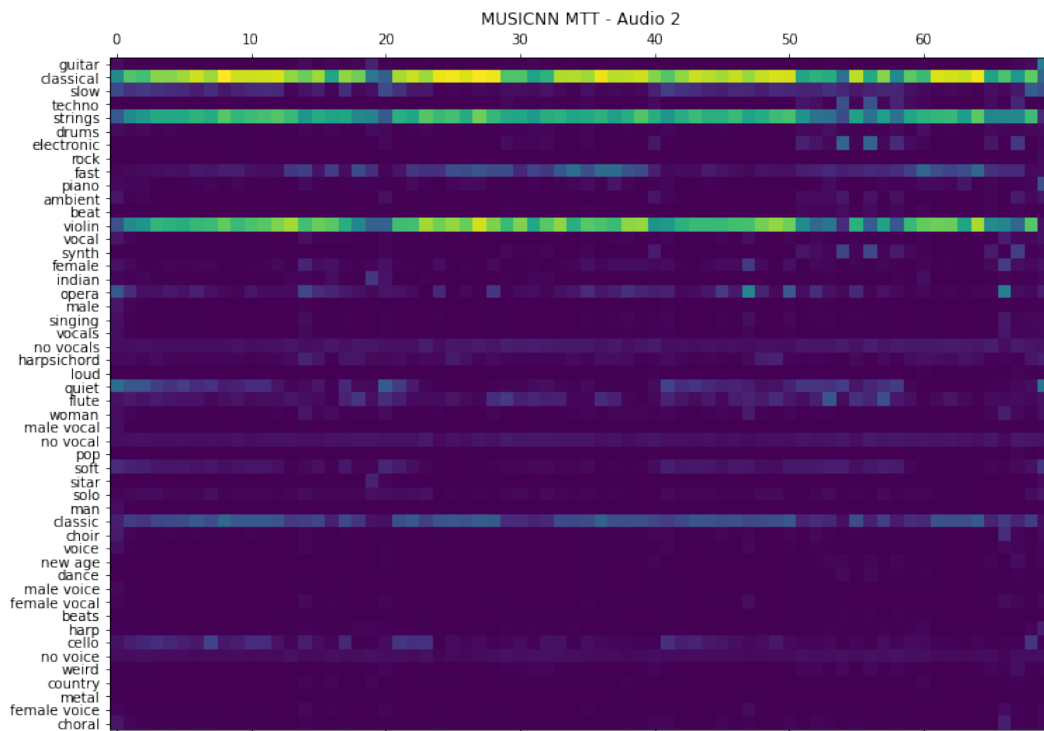


Figure 36. MusicNN MTT, audio 2



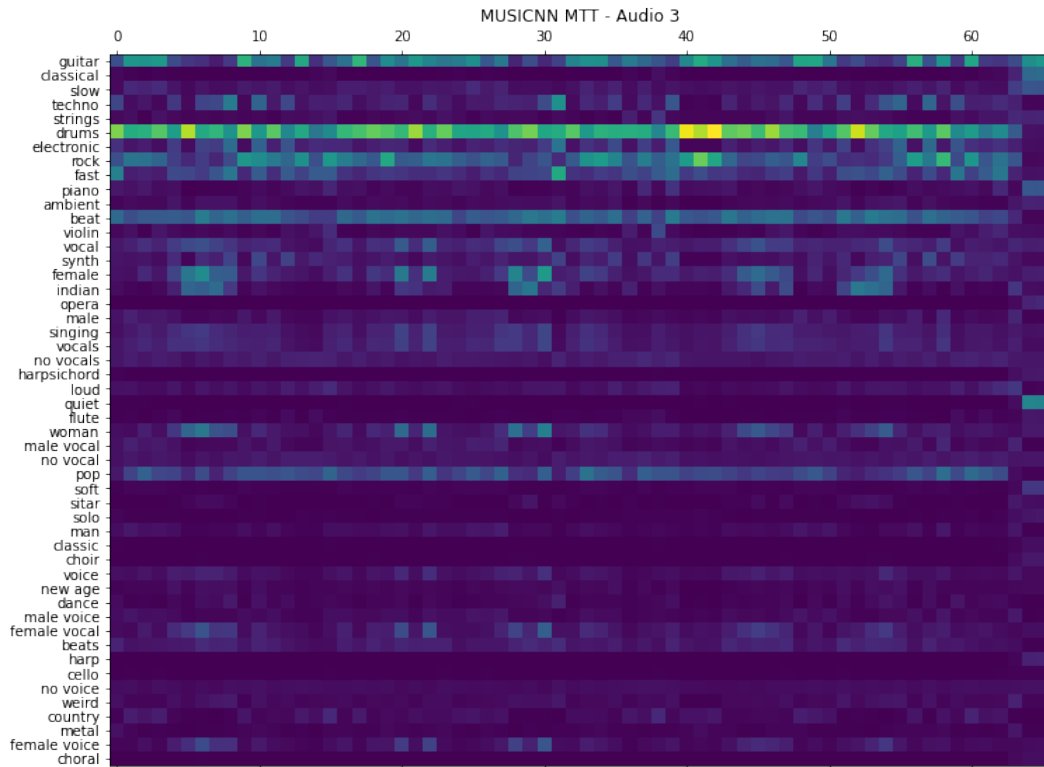


Figure 37. MusiCNN MTT, audio 3

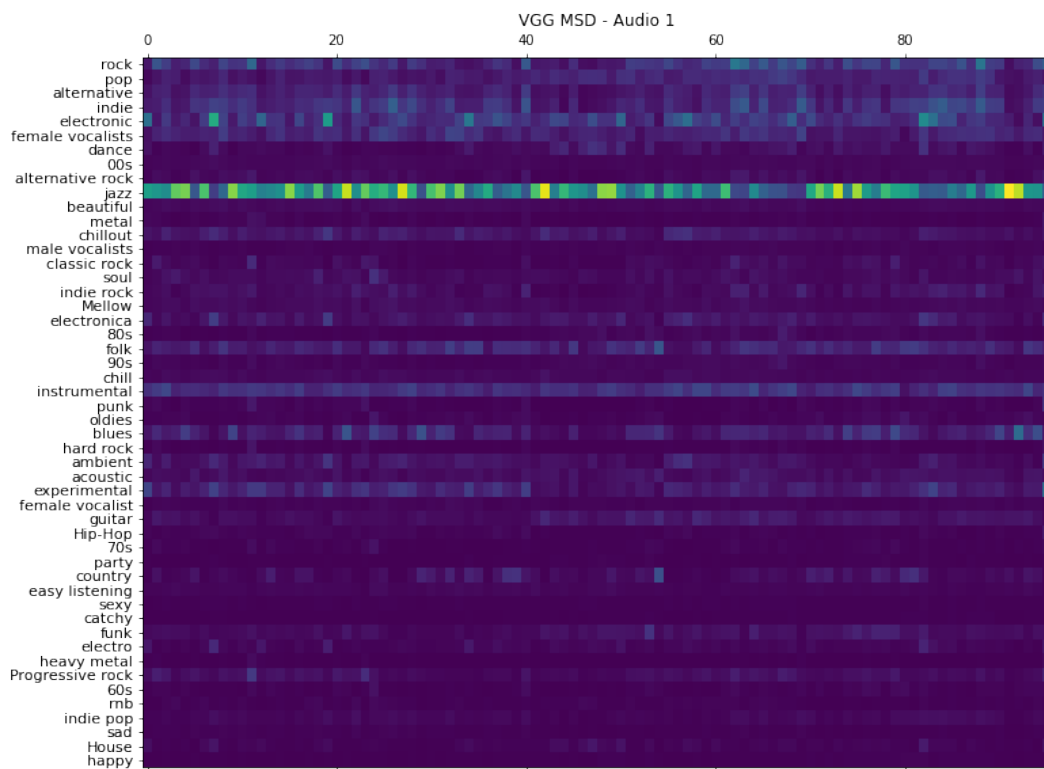


Figure 38. VGG MSD, audio 1

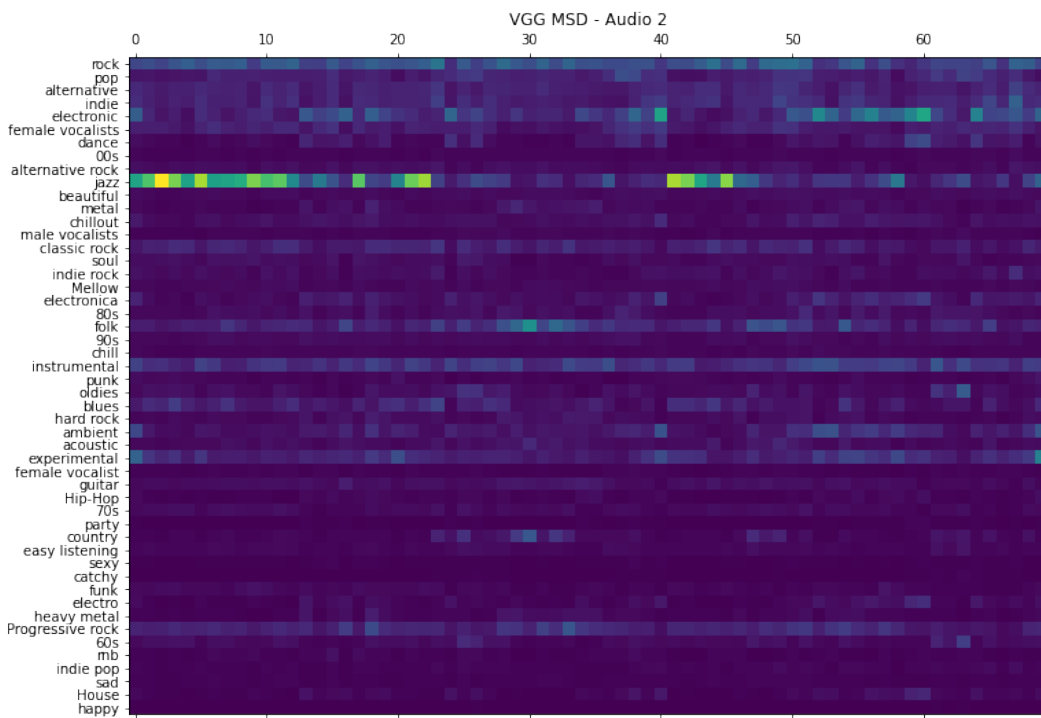


Figure 39. VGG MSD, audio 2

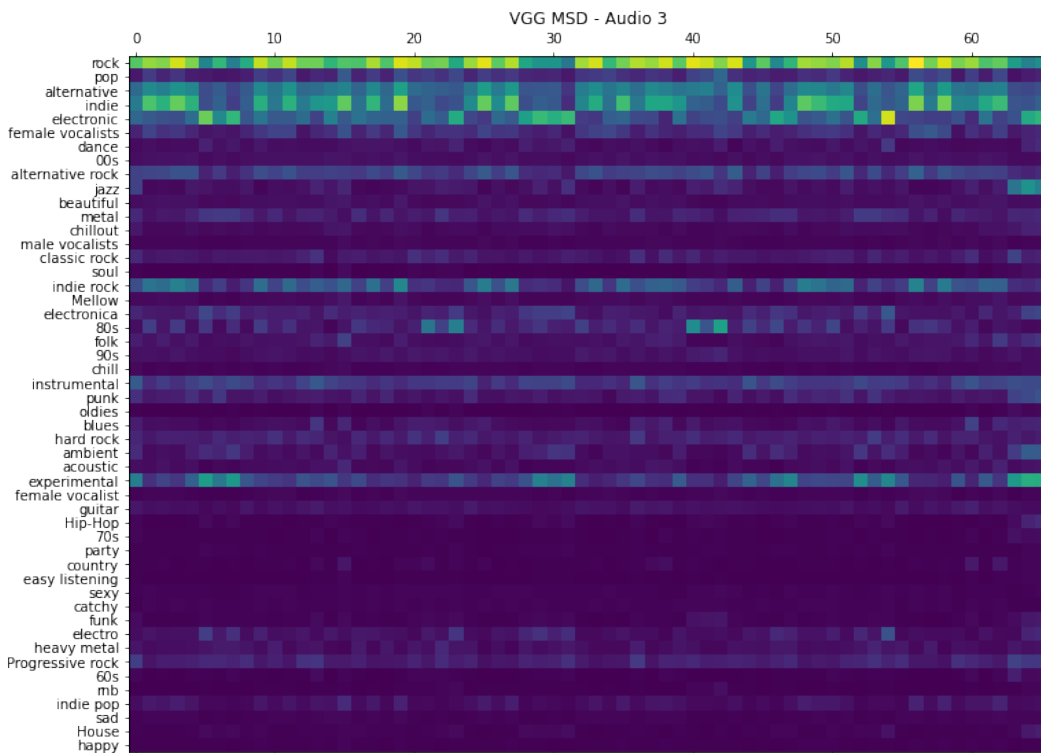


Figure 40. VGG MSD, audio 3

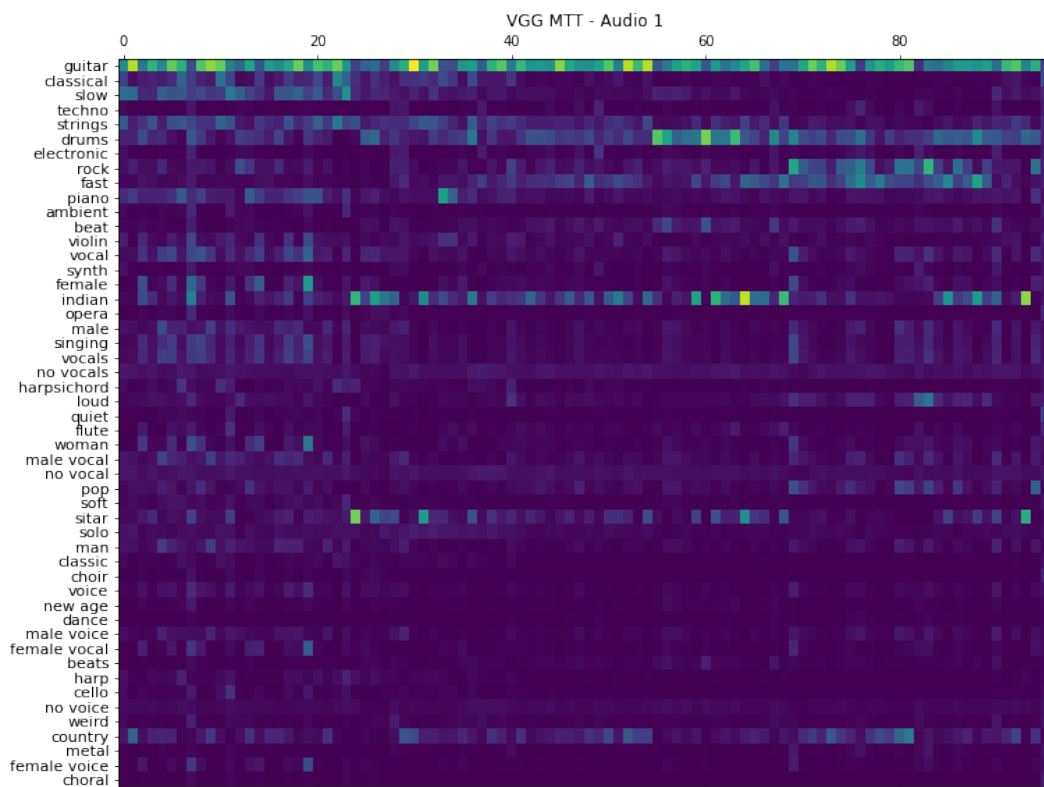


Figure 41. VGG MTT, audio 1

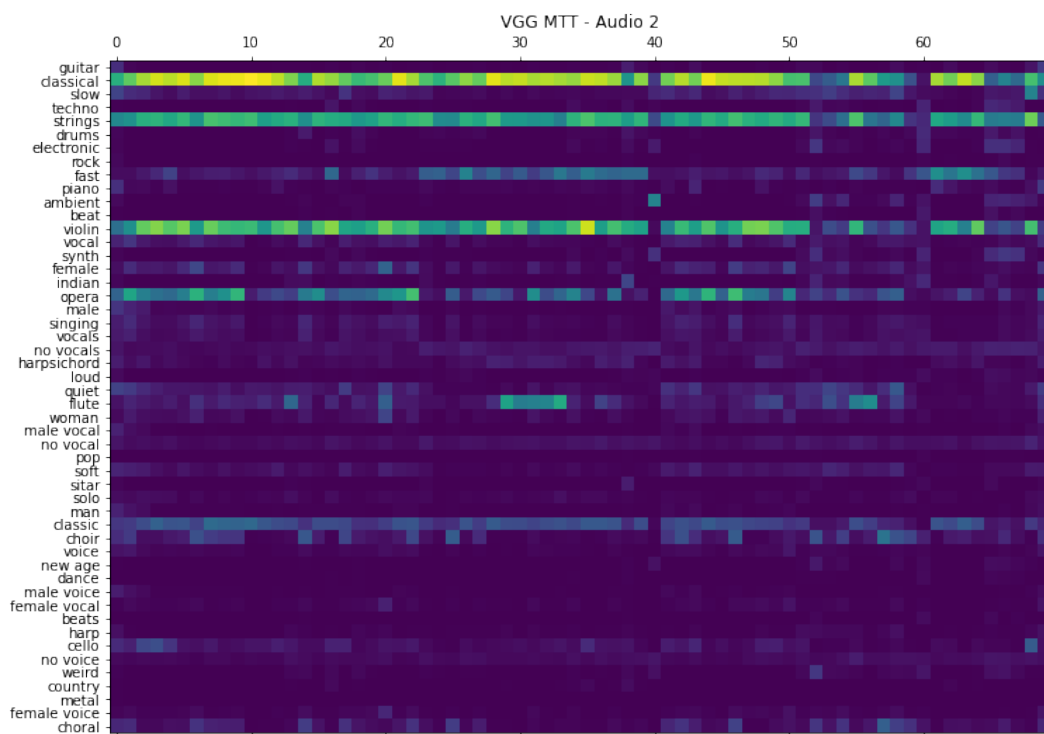


Figure 42. VGG MTT, audio 2

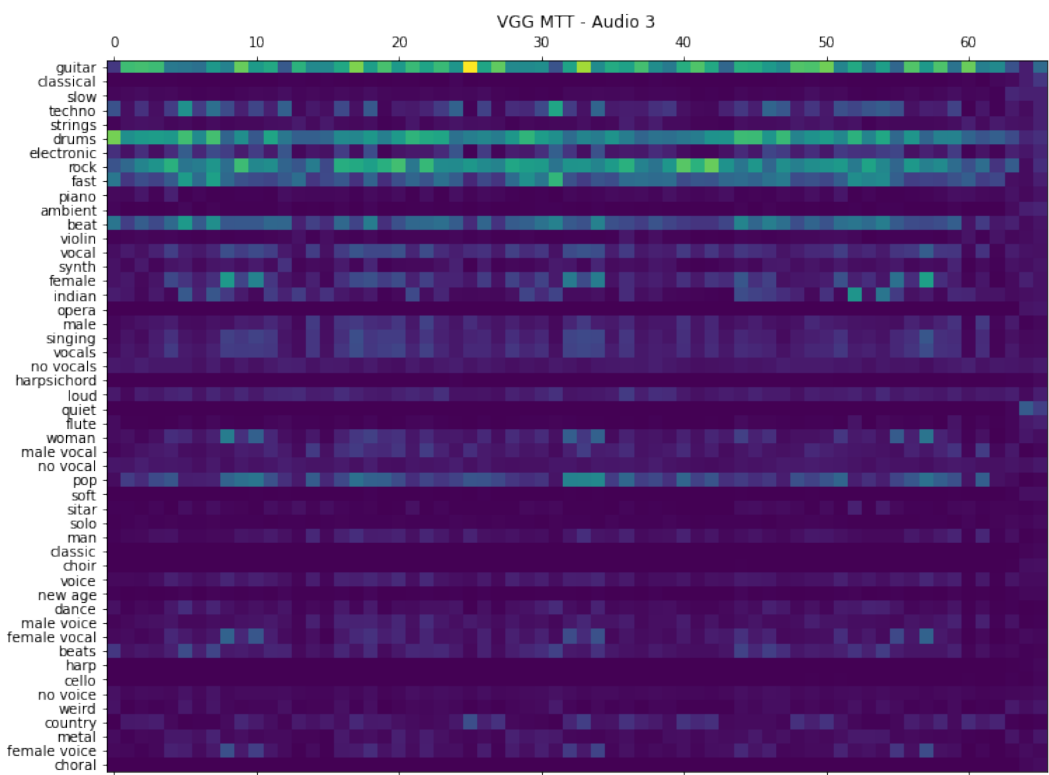


Figure 43. VGG MTT, audio 3