University of Sunderland

PROM02 - Computing Master's Project (2022/2023)

Melody to Symphony: Music Generation from Humming

IBOURK Youssef

MODULE LEADER: Dr Neil Eliot

Table of Contents:

CHAPTER 1: INTRODUCTION

Abstract

In the realm of music, where individual tastes abound, the landscape continues to expand. The creation of new songs is a daily occurrence, captivating the interests of people from diverse cultures and nations worldwide. Our research aims to delve into the fusion of music and artificial intelligence (A.I.).

In recent times, deep learning has made significant strides in generating text through techniques such as natural language processing (NLP). However, the field of music generation remains relatively unexplored. While some projects have explored the generation of new music, our focus is on investigating the feasibility of music generation utilizing existing language model architectures. Specifically, our project centers around the generation of music, where we aim to model musical data similar to human language.

The development of machine learning and deep learning technologies has led to significant refinement and improvement in these systems over time. They have shown remarkable accuracy in generating songs. The challenge of music generation from singing is still difficult to solve. This difficulty arises from the fact that people's humming sounds quite different from one another and from the original music in terms of pitch, pace, and rhythm. Despite these obstacles, there is a lot to gain by perfecting a human humming-based music generation system.

A user of such a system may be able to finally put a name to the tune playing in their minds. It might help musicians create new melodies against the vast musical archive to ensure they aren't infringing on anybody else's work. This technology might inspire the creation of entertaining, engaging applications, giving users new ways to engage with music. Furthermore, it offers a fascinating chance to investigate the limits of existing artificial intelligence tools. Therefore, the goal of this research is to find a way around these obstacles and create a complex system that can reliably generate music from hums.

Background

Music is a global language that connects people across borders, generations, and time periods. Giving people a powerful channel through which to share their thoughts, feelings, and experiences (MacDonald et al., 2021). In addition to enhancing happy occasions, music may also ease the pain of tribulations. The significance and pervasiveness of music in our daily lives demonstrate the worth of music generation systems. OpenAI's MuseNet and Magenta, two of the first music creation systems, fundamentally altered the way in which people engage with recorded. These systems, which rely heavily on audio fingerprinting technology, demonstrated the potential of AI in creating original compositions and expanding the boundaries of music creation.

Creating a music generating system by humming presents a number of obstacles and difficulties. One significant challenge is converting humming or speech recordings into useful musical data in MIDI format. Pitch, rhythm, and other musical components must be extracted from raw audio using advanced signal processing and machine learning techniques. Furthermore, due to the lack of unambiguous notational information, deciphering the intended musical themes from humming is difficult. It can be difficult to precisely extract the underlying musical semantics. Furthermore, humming is subjective and varies greatly between persons, resulting in uncertainty and ambiguity in collecting and conveying hummed data while retaining its essence. High-quality training datasets are required for creating successful machine learning models for music creation, which can be time-consuming and resource intensive. Finally, harmonizing a hummed melody involves generating suitable chord progressions and accompaniment, requiring sophisticated algorithms and music theory knowledge.

Despite these obstacles, there are strong reasons to support the creation of such a system. For starters, it fosters creativity and accessibility by allowing people without professional music training to express their musical ideas. It provides a simple interface for everyone to engage in music composition and innovation. Second, because humming captures the genuine subtleties of human singing, the technology keeps the expressiveness and emotional connection of the original performance. This preserves the created music's personal touch and emotional depth. Finally, humming is an effective strategy for producing musical ideas and motivating songwriters. Musicians can swiftly capture melodies or rhythms that come to mind and experiment with other musical possibilities. Lastly, a humming-based system facilitates collaboration among musicians, enabling easiest way of sharing and exchanging their musical ideas. It provides a platform for co-creating music and fostering a collaborative music-making community.

Because of this that the impact of this system goes beyond the development phase. It has the potential to boost musical creativity by encouraging experimentation, improvisation, and the discovery of new musical terrain. Furthermore, the system may be used as a teaching tool for music theory, composition, and improvisation. Learners may develop a practical and engaging grasp of the link between melody, harmony, and rhythm by participating in this activity. Finally, the method assists music producers and sound designers in producing first ideas or prototyping musical pieces.

Research Aim

The main goal of this study is to create a music generation system based on deep learning that can generate songs based on hums. To do this, we will convert hummed melodies into understandable data, capturing pitch and rhythm information from vocal recordings from which we will interpret and analyze the hummed input, extract meaningful musical elements and semantic information. The algorithm then generates harmonizations, accompaniments, and compositions based on the hummed melodies.

Research Questions

- What deep learning algorithms work best for music generation from humming?
- How can we efficiently represent and preprocess audio data to expand the performance of the model?
- Which humming characteristics should be removed in order to generate songs accurately?
- How can we make sure the system is simple to use and operates well in real-time?

Research Objectives

- To research the top deep learning models for tasks involving audio gratitude and choose the best model for our system.
- To teach the deep learning model how to take the most important musical elements out of the input of humming.
- To provide a real-time interface for recording and dispensation humming input.

Evolution of Music Composing Systems

Since its inception, emerging technologies have influenced the world of music. From the invention of records through the electronic revolution in music, and now to AI technology, the influence of technology in the artistic process is expanding. Many recent achievements in the music business are being fueled by decades of study and discovery in AI technology. In this post, we will examine the progress of AI music approaches to discover how this technology can generate new sounds, compose melodies, build full compositions, and even mimic human singing.

Since the 1950s, the use of artificial intelligence in both understanding and generating music has expanded substantially. The huge growth in AI music intelligence from primitive algorithms to a multi-faceted market with intelligent-music systems demonstrates a technological extension of AI techniques

First period from 1950 to 1970

The earliest computer-generated music attempts occurred in the 1950s, with an emphasis on algorithmic music composition. The introduction of computer-generated music by pioneers such as Alan Turing with the Manchester Mark II computer opened up new opportunities for study into music intelligence, where computational systems could recognise, create, and analyse music.

Early attempts focused on algorithmic composition. The first music produced entirely by artificial intelligence — Illiac Suite for String Quartet — is published in 1957.

Lejaren Hiller (American composer) and Leonard Isaacson (American composer and mathematician) created Illiac Suite, the first original music produced by a computer, using mathematical models and algorithms. They accomplished this feat by employing a Monte Carlo method, which generates random numbers that correlate to musical qualities like as pitch or rhythm. These random aspects were confined to those that would be musically 'legal' as specified by standard musical theory norms, statistical probabilities (such as Markov Chains), and the two composers' creativity.

Iannis Xenakis, a musician and engineer who exploited stochastic probabilities to help his music creation, was another pioneer in this discipline. A stochastic process is a system that has random probability distributions and cannot be anticipated but may be quantitatively analysed. In the early 1960s, he used computers and the FORTRAN programming language to interweave various

probability functions to determine a composition's general structure and other elements (such as pitch and dynamics).

Xenakis created his music as if it were a scientific experiment. Each instrument, like a molecule, went through its own stochastic, random process to establish its behaviour (pitch frequency and velocity of specific notes).

Transitional Period from 1980 to 1990

The emphasis shifted from simpler, algorithmic generation to generative modelling in the decades preceding the current age of music. Otto Laske, a notable Sonology researcher, sees this shift as the difference between a musical robot and musical intelligence.

Musical robot is more like the early efforts of the 1950s and 1960s, it can has a grammar for music, recognise patterns, and has a broad understanding of problem-solving, but it accomplishes its goals in a pretty straightforward and brutal manner. Musical intelligence, on the other hand, substitutes the robot's brute-force searching strategy with a knowledge-based understanding system that is aware of how musical parts may operate.

This tendency of AI systems developing their own self-sufficient understanding of musical aspects served as the foundation for today's higher-level music intelligence.

David Cope (a composer and music professor) was a strong believer in the 1980s that the reach of computer composition might encompass a greater grasp of music through his three primary methods:

- deconstruction (analysis and segmentation);
- signatures (commonalities — keeping what denotes style);
- compatibility (recombinancy – the process of recombining musical parts to create new compositions).

His work centred around the concept of recombinancy, which involves combining and modifying aspects from prior works to produce new pieces of music. Some of the greatest composers of all time deliberately experimented with recombinancy as they transformed previous ideas/styles into their own work. David Cope's goal with EMI was to reproduce this behaviour using computers and their computational power.

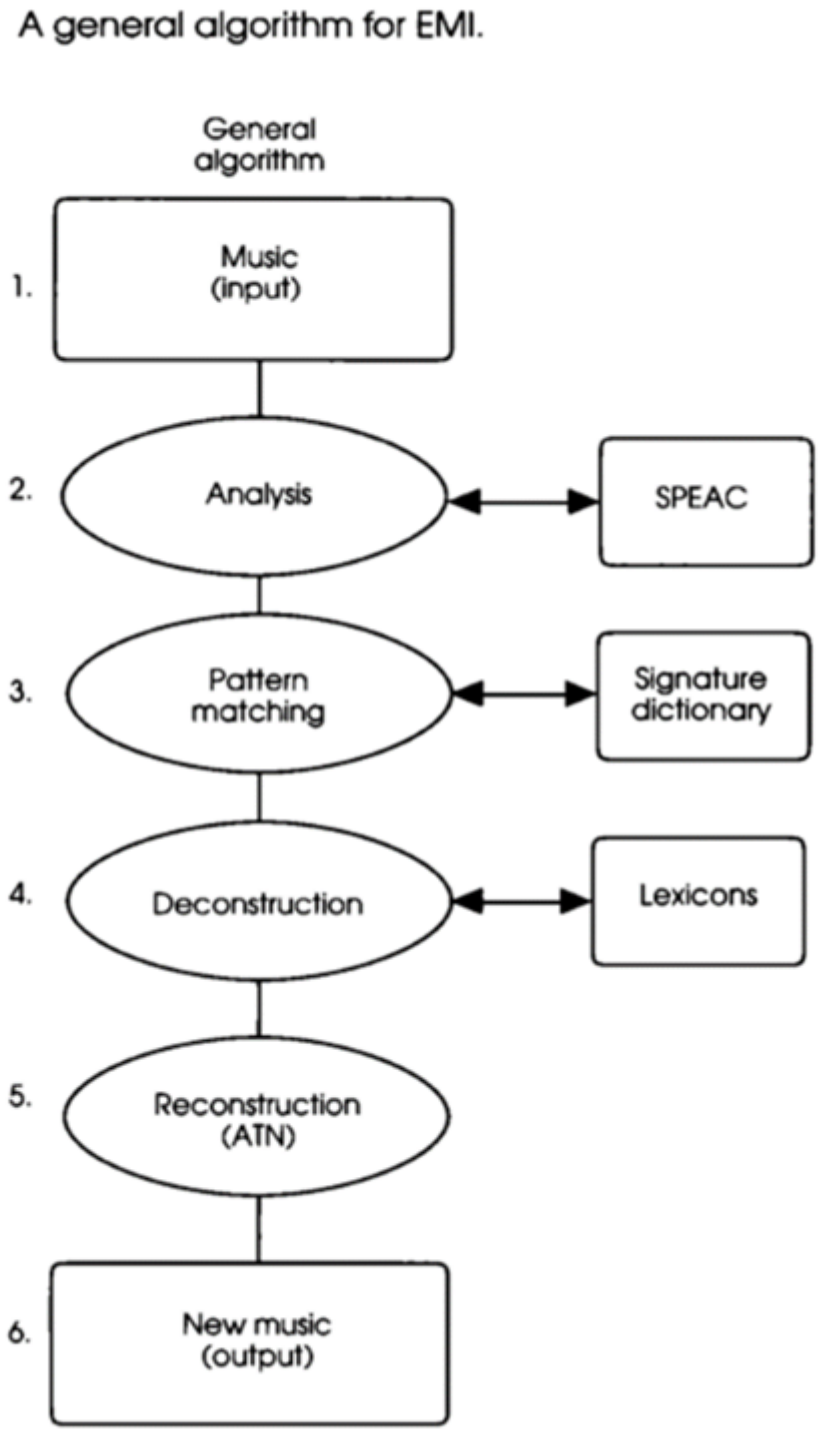


Figure 1: Cope's Six-Step EMI Process. Experiments in Music Intelligence image by David Cope.

Cope's work served as the foundation for several modern AI models on the market. After encoding music and its properties into databases, the collection of recombinant segments is retrieved using specific IDs and pattern matching tools. Then, utilising augmented transition networks, musical parts are categorised and rebuilt in a logical, musical sequence until new music output is created. This form of'regenerative' music creation is reminiscent of many of today's neural networks that generate music.

Other breakthroughs during this time period continued to push the limits of computational creativity. For example, Robert Rowe developed a technique that allows a machine to deduce metre, speed, and note lengths while a person plays freely on a keyboard. Imagination Engines also trained a neural network with popular tunes in 1995 by activating reinforcement learning, resulting in the creation of over 10,000 new musical choruses. Reinforcement learning is the process of training a neural network to achieve a goal by rewarding or punishing the model based on the decisions it makes in order to attain a certain objective.

Current Period (2000s - Present)

The roots of algorithmic composition and generative modelling have dynamically spread into higher-level research as well as into the music industry in the current era of music AI technology. The importance of AI music intelligence in the creative process has expanded dramatically with the adoption of more experimental algorithms and deeper neural networks.

Iamus Intelligent Composition

Melomics' Iamus project was the first to compose classical music in its own style in 2010. Iamus is a computer cluster that composes musical pieces using evolutionary algorithms, which differs from Cope's generative modelling based on existing music.

A randomly produced piece of music is modified (adjusted in pitch, dynamics, etc.) and subjected to a set of rules to determine if it complies with genre or music theory conventions, much like the process of natural selection. After only a few minutes, this development enables a random input fragment to transform into hundreds of pieces that adhere to true music requirements.

Magenta Music Analysis

Magenta is a Google Brain initiative that employs machine learning to assist the creative process. They have developed a variety of applications that demonstrate some of music intelligence's capabilities, such as transcribing audio using neural networks or merging musical scores utilising what are known as latent space models. However, via their studies with MusicVAE, we can observe the breadth of Magenta's music analysis.

MusicVAE is a machine learning model that can combine musical scores to create new compositions. They do this by employing a latent space model, which converts higher-dimensional dataset variance into a more accessible mathematical language. This is accomplished by the use of an autoencoder, which takes a collection of melodies and compresses (encodes) each sample into a vector representation, which is subsequently reshaped into the same melody (decoding).

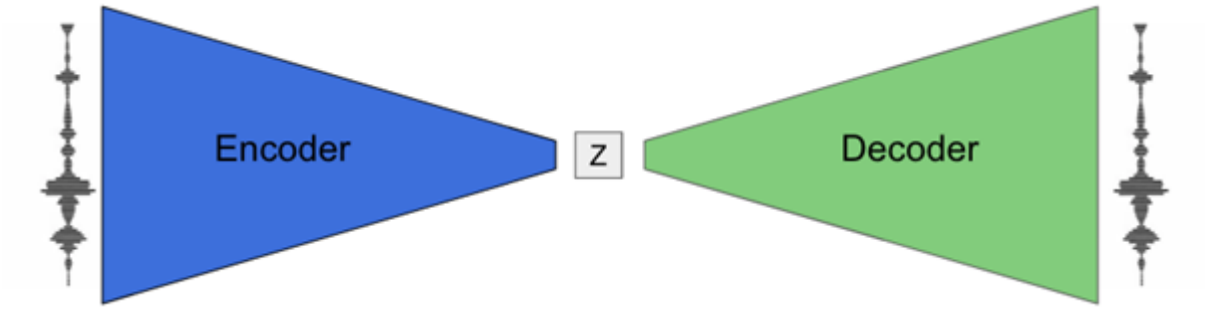


Figure 2: Magenta's autoencoder compressing and decoding audio. Image by Magneta

The autoencoder learns the properties that are comparable throughout the whole dataset after learning how to compress and decompress numerous inputs. MusicVAE is based on this premise, but it incorporates hierarchical elements to build a long-term framework.

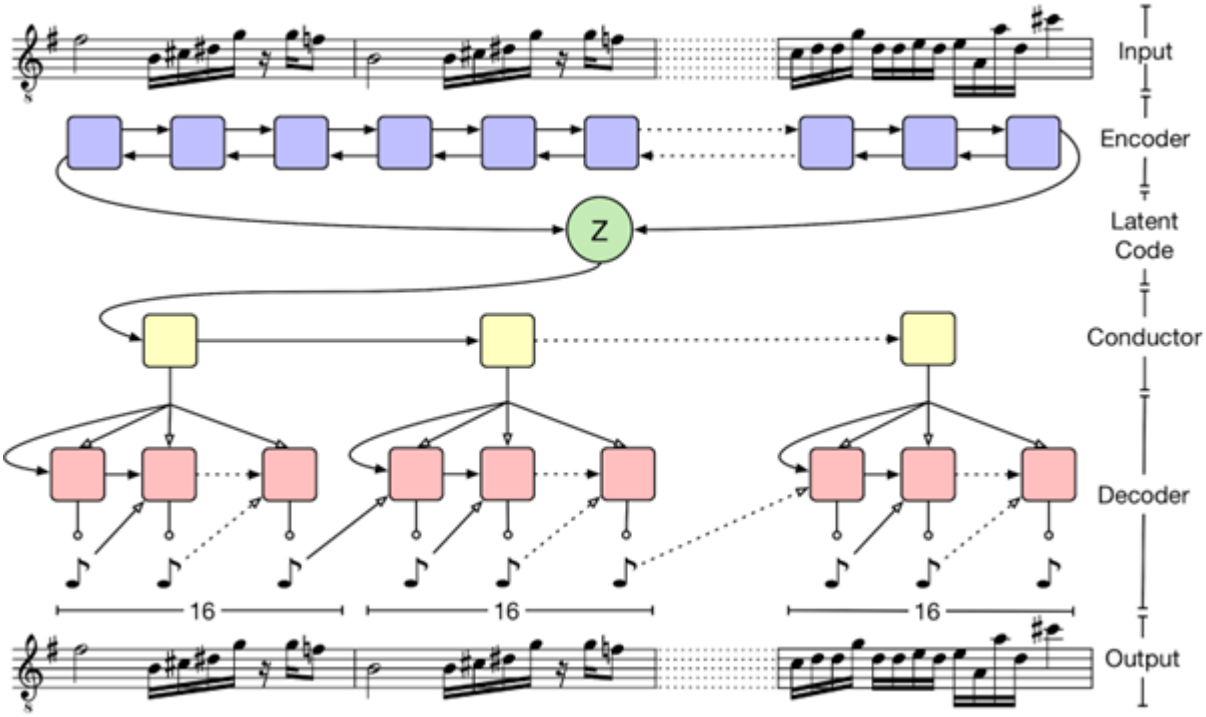


Figure 3: The hierarchal autoencoder structure. Image by Magneta

MusicVAE may build many apps based on this framework to create interpolations, percussion patterns, and whole new melodic loops depending on an input of your music.

Based on this architecture, MusicVAE had the ability to create a variety of apps that generate interpolations, percussion patterns, and completely original melodic loops from the input of your music.

Sound Creation Using NSynth

While the majority of early research concentrated on the act of composition, recent research has

evolved to encompass machine learning in the field of sound synthesis. Magenta's NSynth (Neural Synthesiser) creates sounds at the level of individual samples rather than oscillators/wavetables like standard synthesisers do. This method provides more artistic control over timbre (the characteristic quality of a sound) to aid in the creative process.

On a dataset of more than 300,000 musical notes from more than 1,000 instruments, NSynth employs a Wave-Net type autoencoder. Based on the total probability theorem, this one-of-a-kind dataset allows for the factorization of a musical sound into notes and other properties.

Jukebox's Generative Modelling

Most attempts at automatically creating music, often using a piano roll or MIDI, which act as languages for describing sounds and sequences. With its direct modelling of audio such as music and human speech, OpenAI's Jukebox advances generative modelling. Jukebox's methodology enables them to produce melodies, compositions, timbres, and even basic vocals in a range of genres and styles. They make use of neural networks and specialised encoders to address the depth of the semantic information in raw audio.

Jukebox used VQ-VAE encoders to compress its audio into a latent space. The total system begins by encoding audio with Convolutional Neural Networks and then detecting patterns in these encodings. A convolutional neural network (CNN) is an algorithm that can take an input picture and mathematically express its unique properties using multidimensional matrices. CNNs may be used to audio by using a visual representation of an audio sample (such as a spectrogram).

The advancement of music intelligence technology has resulted in a plethora of new commercial applications. LANDR, for example, employs deep learning algorithms for automated audio mastering. Furthermore, numerous products have used a mix of neural networks and reinforcement learning algorithms to develop commercial tracks, such as Taryn Southern's album (co-produced by Amper Music) and the Hello World album (created with Sony's Flow Machines and musicians). We will witness more AI-assisted music in popular contexts in the near future as these algorithms and neural networks mature

The Power Of Humming

Because humans have an innate capacity to replicate sounds, humming is frequently utilised as a substitute for instruments or lyrics. We hum tunes that have been ingrained in our heads, favourite songs, and even make our own songs. Humming has exceptional importance since it can be done by anybody, everywhere, regardless of language or culture (Meyer and Moore, 2021). This

seemingly simple human behaviour, however, presents an interesting and challenging task for music generating algorithms.

According to anthropologists, humming is a pre-linguistic mode of human communication. For millennia, humans have used music to communicate sentiments, speed, and melody. Nowadays, it's still a natural, spontaneous method to express yourself and let out your sentiments. Humming is a natural response when we are happy, introspective, or worried. Humming is an essentially personal and well documented way of expression because to its intimate link with our moods. Humming, from a musical standpoint, represents the melody of a song without the accompaniment of singers or instruments. Making a song out of a hummed tune is a strong and touching testament to the value of melody in music. Even if we've forgotten the words or other details, we can generally sing or whistle the music. Because it is both widespread and simple to analyse, humming is a fascinating issue area for music generation technologies.

Despite its apparent simplicity, humming adds a lot of complexities to music generation algorithms. The broad variation in how humming is performed is a big impediment. Hummed music's pitch, rhythm, speed, and timbre may vary from person to person. Individual variances in musical aptitude, song memory, emotional state, and cultural background all have the potential to increase this type of diversity. Humming, on the other hand, is typically done without the harmonies and accompaniment of a full musical arrangement. Without these auxiliary musical pieces, we are left with a simple melody line that can be interpreted in a variety of ways (Ji et al., 2020). Because of the time nature of music and the abstractions involved, humming adds still another degree of complication. When faced with such complications, older music generation systems and traditional machine learning algorithms generally fall short.

The capacity of deep learning to detect patterns in high-dimensional data makes it an appealing option to overcome these challenges. Music creation has gone a long way in recent years, but it still requires a thorough understanding of music theory, exact feature extraction, and effective handling of the inherent instability of human humming. Music creation algorithms must be able to accommodate humming as well as any other sort of musical expression due to their intricacy. Humming analysis has the ability to enhance not just music production but also how we react to music in its most basic form. The ultimate goal is to develop a system that combines the ubiquitous nature of humming with the pinpoint accuracy of machine learning, further merging human experience with the power of AI.

CHAPTER 2: LITERATURE REVIEW

Universality and Simplicity of Humming

Osikominu and Bocken, 2020 dicuss that humming is a universal human activity that may be seen in a wide variety of countries and cultures. For folks who can't play an instrument or are singing a melody they've forgotten, this is often their only means of expressing themselves musically. This commonality has its roots in the innate human ability to recognize and imitate patterns. Our minds are wired to pick up on and repeat memorable melodies, and when we hear a tune we like, we frequently find ourselves humming the tune (Osikominu and Bocken, 2020).

Humming's ease comes from how readily you can do it. Humming a melody requires neither a high level of musical ability nor an in-depth knowledge of music theory. The music is reduced to its most elemental form, the melody, by the lack of words and intricate musical arrangements. This makes humming a very intimate and expressive method of communication. When we hum, we are communicating more than just the sounds we hear; we are also sharing the emotion or memory that comes with that song. Since humming connects our internal emotional world with the exterior musical world, it may be seen as a bridge between the two.

The Complexity of Humming for Music Generation Systems

Eremenko et al., 2020 argue that humming, although being so ubiquitous and easy to do, offers a number of challenges when it comes to actual music creation. To begin, there is the problem of variation. No two people can hum the same music in sync (Eremenko et al., 2020). Humming may vary in pitch, rhythm, and pace depending on the hummer's musical talent, recollection of the song, personal style, cultural background, and even emotional condition at the moment of humming. A successful music generation system, therefore, has to be strong enough to recognize the fundamental melody despite this inherent unpredictability.

Second, compared to listening to a complete recording of a song, humming doesn't provide nearly as much musical evidence. The melody line is isolated from the song's lyrics, harmony, instruments,

and production. Because of this, the music generation system must be very adept at picking up patterns and elements in the music in order to succeed. Finally, humming adds an extra layer of abstraction, which makes music generation even more difficult. Simply said, when someone hums a tune, they are offering their own unique take on the tune based on their own unique set of perceptions and memories.

These challenges highlight the need for a sophisticated combination of cutting-edge audio processing techniques, deep learning algorithms, and robust feature extraction methods in a humming-based music detection system. (Eremenko et al., 2020) A system of this kind would have to do more than just identify the main melody of the hummed song; it would also have to understand the nuances and abstractions that the human performer would inevitably bring. Because of this, humming-based music detection is an interesting and difficult subject in the larger area of Music Information Retrieval (MIR).

Role of Machine Learning in Music Composition

The success of machine learning in computer vision and speech recognition has driven scientists and researchers to apply similar approaches to music creation. The important areas that require additional investigation are music composition, analysis, and suggestions. The major difficulties under emphasis are digital audio signals, processing, and modelling of an effective machine learning system.

Machine learning, which employs algorithms that learn from data, enables computers to make predictions, recognise patterns, and adapt to new information without being explicitly programmed. Machine learning has made tremendous advances in audio processing and music synthesis, enabling the development of previously unthinkable tools and applications.

Speech recognition is one of the most noteworthy uses of machine learning in audio processing. As we communicate with voice assistants such as Siri, Alexa, and Google Assistant, this technology has become a vital part of our everyday life. These voice assistants can now manage gadgets, search for information, and connect with people using only their voices, thanks to machine learning algorithms. Not only has this enhanced the user experience, but it has also made technology more accessible to those with impairments.

Machine learning has also been utilised to create complex audio processing systems that can improve sound recording quality. Noise reduction algorithms, for example, may be trained to recognise and eliminate undesired background noise from audio files, resulting in cleaner and

clearer recordings. Machine learning may also be used to create systems that automatically alter the volumes of various audio components, such as vocals and instruments, to ensure a balanced mix. These developments have benefited musicians and audio engineers in particular, who can now obtain professional-quality results with more ease and efficiency.

Machine learning has also been used in the creation of music in recent years. Researchers and musicians have been experimenting with algorithms that are capable of creating creative music in a variety of styles and genres. Google's Magenta project, which employs machine learning to generate music and art, is one such example. Magenta's algorithms can analyse current music, understand its structure and patterns, and then create new songs based on that information. This has resulted in the composition of totally new works of music that are both unique and artistically coherent with the ones that inspired them.

While some would contend that the use of machine learning to the creation of music endangers the importance of human creativity, others think it might be a useful tool for musicians. Composers and musicians might discover inspiration and pursue previously unconsidered creative directions by employing algorithms to produce fresh musical ideas. Machine learning may also be used to create compositional tools that help artists focus on the more emotive parts of their work by producing chord progressions or recommending melodic ideas.

With a wide range of applications, machine learning's influence on audio processing and music production is fast growing and changing how we listen to and make music. Machine learning is allowing the creation of novel technologies that are reshaping the audio business, from voice recognition and music recommendation to advanced audio processing tools and music synthesis. It will be exciting to observe how machine learning further affects the future of music and audio processing as these developments progress.

Deep Learning: The Game Changer in Audio Processing

Deep learning has now established itself as the main paradigm for modelling and processing audio signals. While the phrase "deep learning" may refer to any technique that does deep processing, we describe it here as a deep stack of non-linear projections created by non-linearly linking layers of neurons; a process reminiscent of biological neural networks.

Deep Learning comprises a diverse range of designs and training procedures that differentiate how neurons are linked to one another, how spatial or temporal information is considered, which criteria are being optimised, and for which purpose the network is intended to be employed.

Different characteristics of speech, music, and ambient.

Music and ambient sound audio signals are composed of several simultaneous sources, whereas spoken audio signals typically have a single source. Some musical instruments, such as the piano, can create many tones at once because they are polyphonic. This makes it particularly difficult to analyse musical and ambient sounds. When using the rules for time-frequency representations of audio signals, speech is highly organised through time. This structure results from the usage of a language-specific vocabulary and grammar. Additionally, music is very well-structured both vertically (many simultaneous sound occurrences) and horizontally (across time).

Music processing Deep Learning

Music processing is connected with Music Information academic (MIR), an interdisciplinary academic subject. This discipline is concerned with the comprehension, processing, and creation of music. It blends music theory, computer science, signal processing, perception, and cognitive ideas, concepts, and techniques. MIR is concerned with the development of algorithms for:

- *summarising the music's substance* based on an examination of its audio signal. Examples of this include estimating different pitches, chords, and rhythms; identifying the instruments used in a piece of music; assigning "tags" to a piece of music (such as genres, moods, or usage) allowing for the recommendation of music from catalogues; and detecting cover/plagiarism in catalogues or in user-generated contents.
- *analysing the music's substance*. Source separation and enhancement are examples of this.
- *creating new audio signals* or musical compositions, or converting one signal's qualities into another.

Environmental sound processing Deep Learning

Environmental sound processing is related to the research area known as Detection and Classification of Acoustic Scenes and Events, which focuses on the creation of algorithms for:

- classifying acoustic scenes (determining where a recording was made, such as in a station, office, or street);
- detecting sound events (determining which events occur over time in an audio scene, such as a phone ringing, a dog barking, or a motorcycle passing);
- locating sound sources in space.

CHAPTER 3: Music Generation: Dataset and Implementation

Dataset

The goal is to use machine learning to develop new avenues of human expression, new creative tool for musicians.

To achieve this goal, we will use the NSynth dataset, which is a massive collection of annotated musical notes taken from different instruments at various pitches and velocities. It is an order of magnitude larger than comparable public datasets, with more than 300.000 notes each with a unique pitch, timbre, and envelope. from more than 1.000 instruments having as attributes notes qualities, instrument source and instrument family (bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth lead, and vocal). The richness and diversity of this dataset make it an ideal resource for training our music generation system.

This dataset offers a standard testing ground for consistent qualitative and quantitative evaluation of generative models, despite the fact that it is notoriously difficult to evaluate (Theis et al., 2015).

We were aware of the need for an accessible audio dataset. Many previous attempts at data-driven audio synthesis have focused on more constrained domains such as texture synthesis or training small parametric models (Sarroff & Casey, 2014; McDermott et al., 2009). Audio signals found in the wild contain multi-scale dependencies that prove particularly difficult to model (Raffel, 2016; Bertin-Mahieux et al., 2011; King et al., 2008; Thickstun et al.)

WaveNet is a potent generative method for probabilistic modelling of raw audio (van den Oord et al., 2016a). In this part, we discuss our innovative WaveNet autoencoder structure. The major aim for this strategy is to achieve long-term consistency without the need of external conditioning. A further aim is to put the learnt encodings to use in applications like meaningful audio interpolation.

Based on a combination of human review and heuristic algorithms, we added tagged each note with three extra pieces of information:

- Source: The technique of sound generation for the note's instrument. This can be 'acoustic' or 'electronic' for instruments recorded from acoustic or electronic instruments, respectively, or 'synthetic' for instruments synthesised.

- Family: The high-level family to which the instrument of the note belongs. Each instrument belongs to just one family.

- Qualities: The sonic characteristics of the note. The entire list of classes and their co-occurrences may be found in the Appendix. Each note is labelled with one or more attributes.

However, testing generative models with audio datasets such as NSynth presents distinct obstacles. Audio signals found in the wild have multi-scale relationships and complicated structures, which are notoriously difficult to adequately represent. In this situation, ensuring consistent qualitative and quantitative evaluation becomes critical. Although the assessment process can be difficult, we want to overcome this by assessing the performance and quality of our music generating system using a combination of human review and heuristic algorithms.

To inform our approach and gain insights from existing research, we draw upon literature related to the WaveNet autoencoder structure. WaveNet has showed promise in maintaining long-term consistency without external conditioning by demonstrating its effectiveness in probabilistic modelling of raw audio. This knowledge will be used to create an innovative WaveNet-based architecture for our music creation system, with an emphasis on meaningful audio interpolation and capturing the expressive subtleties of human humming.

Our project aims to develop a music generation system that empowers musicians to express their creativity through humming and provides a powerful tool for exploring new musical possibilities by leveraging the unique characteristics of the NSynth dataset, addressing evaluation challenges, and incorporating insights from the WaveNet literature.

Importance of Feature Extraction

Jaishankar et al., 2023 discuss that the process of feature extraction is crucial to music creation because it simplifies otherwise overwhelming audio information. Therefore, feature extraction plays a crucial role in streamlining this information and zeroing down on the crucial characteristics that constitute a song or piece of music. A compact representation of a piece of music's distinctive characteristics may be formed using the extracted features, which aids in capturing the music's

spirit (Jaishankar et al., 2023). They serve as contributions for further processes in a music generation system, such as categorization and matching, allowing for more precise and effective generation. Not only are musical aspects essential for creating a song, but they also help in determining the song's genre, mood, and even the instruments used. They have several potential uses, some of which include music therapy, automated music transcription, and recommendation systems.

Techniques for Extracting Musical Features

There are several approaches for extracting musical elements, each with its own set of benefits and uses. Spectral analysis is a typical approach that involves splitting audio signals into their basic frequency components using methods such as Fourier transforms. Pitch, timbre, and spectral energy distribution, for example, give essential information about musical content and may be utilised to record various components of music, ranging from melodic patterns to instrument characteristics. Note pitches may be determined by analyzing the spectral data and locating the prominent frequency components. This pitch data is used as the foundation for melody generation, which often employs methods like Hidden Markov Models (HMMs) or dynamic temporal warping to capture the progression and nuances of notes that make up a melody (Dash and Agres, 2023).

Dash and Agres, 2023 discuss that the melody, or emotional heart, of a song is largely determined by the notes' pitches. Pitch detection uses a number of different techniques, including the YIN algorithm, the Harmonic Product Spectrum (HPS), and the Autocorrelation technique. In the preprocessing stage, these methods include converting the original audio signal to a spectrogram or other frequency domain representation.

Rhythm analysis is another approach that focuses on obtaining temporal characteristics linked to music timing and rhythmic patterns. Beat detection algorithms may be used to recognise a musical piece's rhythmic structure, allowing the development of rhythmic patterns and grooves. Tempo, beat power, and rhythmic intricacy are all examples of rhythm elements that may help create captivating and engaging music.

Harmonic analysis is another important feature extraction approach, notably for music production including chords and harmony. Harmonic characteristics in a musical piece record chord progressions, key signatures, and harmonic connections. Chord recognition, harmonic pitch class profiling, and harmonic change detection can all give useful insights into the harmonic structure of music, allowing for the creation of harmonically coherent and pleasant musical sequences.

Furthermore, structural characteristics such as musical form, segmentation, and phrase boundaries may be extracted using feature extraction approaches. These characteristics aid in capturing the general structure and organisation of a musical composition, allowing for the creation of music with well-defined parts and cohesive musical storylines.

Implementing a Music Generation System

Teney et al., 2020 discuss that a humming-based music generation system's UI is just as important as any other aspect of the program. It acts as the interface between the user and the system, facilitating communication between the two. It is impossible to overstate the value of a simple and straightforward interface, since it has a direct bearing on the quality of the user's experience and, by extension, on the system's level of acceptability and ultimate success. The fundamental goal of the user interface of a music generation system is to make it simple for the user to record their own humming and provide meaningful results. The interface has to provide a natural progression of actions (Teney et al., 2020). Users should be greeted by a warm home page and given clear directions on how to record their humming from there. This may include just pressing a button to begin and end the recording, or it may involve more complex features like visual indications of volume levels and countdown clocks to ensure everything is set up properly. Adding features for immediate feedback to the user interface may improve it even more. These may provide users with real-time, interactive feedback on the quality and loudness of their humming, allowing them to fine-tune their input and get the best conceivable recognition results. Providing users with such immediate responses may greatly increase their involvement and contentment with the system. The user interface plays a significant part in the generation process. User expectations should be managed throughout this processing time by providing some type of progress indicator or other engaging, informative placeholder content. The findings of the generating process should be presented in a clear and visually acceptable way via the interface.

Evaluating the Performance of Music Generation Systems

Metrics for Evaluating System Performance

Kar and Corcoran, 2017 discuss that the foundation of every system design process is a thorough performance review. When evaluating the efficacy of a music generation system, it is important to consider a number of different measures. Accuracy is the most important indicator since it

measures how well the algorithm can generate songs from hummed inputs. Prediction accuracy may be measured by comparing the output of the system to a ground truth dataset of already-established humming-song pairings. The proportion of accurate predictions relative to total predictions serves as the accuracy score. While precision is essential, it isn't enough to provide a whole image, particularly when there is a discrepancy between the song classes. In these cases, a more nuanced understanding of the system's presentation can be gained through the use of additional metrics such as precision, recall and the F1-score (the harmonic mean of precision and recall). Latency, or how long it takes for the system to respond to the input, is another crucial performance parameter (Kar and Corcoran, 2017). A fast turnaround time for results is essential for a successful music generation system. User discontent and even abandonment of the system due to excessive delay is possible. Another key performance metric that has to be assessed is the system's robustness. This includes testing the system's capacity to respond to a wide variety of real-world conditions, such as hearing voices, whining patterns, and degrees of background noise that are all unique. A good music generation system will be able to adapt to changing settings and continue to offer accurate generated song.

User Feedback and Usability Assessment

Barbazza et al., 2021 discuss that User feedback and usability tests give qualitative insights into the system's practical operation that complement the quantitative performance measurements. Such assessments provide a deeper and more actionable insight into how the system communicates with and meets the demands of its end users. To begin, it is essential to assess the system's usability, which includes factors such as accessibility, intuitiveness, and responsiveness. The user should be able to quickly and easily navigate the interface, grasp its purpose, and record their humming input with as little friction as possible. Users' encounters with any difficulties or technological faults throughout this procedure might serve as a pointer of where improvements are needed. Second, the opinions of those who have interacted with the system provide valuable insight about its effectiveness and usability. Users' experiences, opinions on the system's merits and weaknesses, and suggestions for improvement may be obtained via many channels, such as surveys and interviews (Barbazza et al., 2021). Users may be asked for specific comments on things like interface design, recognition speed, and information clarity, or they can be given more freedom to offer general feedback on whatever they'd want. Information may also be gleaned through observing and analysing user actions during testing. This may include how fast users pick up the system, how often they run into problems, and what kinds of patterns emerge in their usage of the system.

Deployment, Integration, and Maintenance

Optimization for Real-Time Performance

Di Mitri, 2021 argue that it is essential to optimize a music composition system for real-time performance so that it can keep up with users' demands for instantaneous responses. The system's latency and processing efficiency may be improved by tweaking a number of its individual parts. The deep learning model itself is a crucial factor that might affect performance in real time. Longer inference times may be the result of using more complex models with more layers and more parameters. Methods like model trimming and quantization may be used to solve this problem. In order to reduce the model's size and processing needs, "pruning" is the process of deleting unnecessary or insignificant connections or neurons (Di Mitri, 2021). Reducing the accuracy of model weights and activations via quantization helps save up storage space and simplify calculations. These methods allow the model to be simplified and its efficiency increased without suffering a notable drop in accuracy. The inference time may be diminished by using efficient computing approaches in addition to optimizing the model. The overhead of individual forecasts may be minimized by the use of batch processing, which involves processing several inputs concurrently. To speed up calculations, GPU acceleration takes use of graphics processing units' inherent parallel processing capabilities. Using specialized hardware like GPUs may greatly accelerate the inference of the deep learning model, enabling real-time performance. Also crucial to real-time performance optimization are the pre- and post-processing phases. Noise suppression, audio feature extraction, and pattern corresponding are all examples of activities that might benefit from the use of more efficient algorithms and methods. Distributing the computing burden over numerous cores or processors through parallel processing may increase the speed at which such processes are performed. Audio processing operations may be made more efficient by the use of hardware acceleration, such as specialized signal processing chips or libraries.

Deployment Platforms and Integration with External Services

List, 2017 discuss that the use case and intended audience will dictate the platform on which a music generation system is deployed. As long as the device has an internet connection, users of a web application may access the system from anywhere. As updates and conservation can be performed centrally on the server, users are not required to download and install any new software. On the other hand, a mobile app takes use of the features of smartphones, such as the in-built microphone and the ability to work without an internet connection, to provide a more

customized and portable experience. Scalability is an important factor to think about regardless of the technology stack. The system has to be prepared for future spikes in demand, so that it can continue to serve a larger audience without slowing down (List, 2017). The use of cloud-based infrastructure and load complementary systems to disperse the computational load over numerous servers are two examples of methods that may be used to achieve scalability. The functionality and convenience of the music generation system may be greatly improved by integration with third-party services. The system may offer extra details about the generated song, such as the song's genre, tonal scale, and chord progression. The connection may also provide users with one-click sharing of the generated music on a streaming service, or a social media platform where they can hear it or add it to their playlist. These services often have their own APIs (Application Programming Interfaces) that may be used to integrate the music generation system with them, allowing for more streamlined communication and data sharing.

System Maintenance and Updates: Importance and Strategies

Shi et al., 2020 discuss that for a music generation system to function reliably and efficiently over time, regular maintenance and upgrades are required. Developers may maintain the system's sustained functionality, fix any problems that emerge, and keep up with the needs of their users and the state of the art in technology by constantly monitoring and updating the system. Monitoring the system's performance on a regular basis is essential for spotting any errors, delays, or other hiccups in its operation (Shi et al., 2020). Logging user activities and system reactions is one way to collect useful information for research. Developers may prevent the system from ever falling short of expectations by keeping an eye on these measures and fixing any problems before they ever become noticeable. Proactive system upgrades are compulsory to include breakthroughs in deep learning and music generation systems, in addition to performance monitoring. Research in the subject is dynamic, thus keeping up with the most recent methods and algorithms may greatly improve the system's efficiency. Updating the deep learning model on a regular basis might include retraining the model with fresh data, tweaking the model's hyperparameters, or implementing whole new methods for extracting and generating audio features. These revisions keep the system up-to-date and useful in a constantly developing industry. In addition, comments from users are essential for the further development of the system. Developers may learn more about possible problems, enhancement opportunities, and user demands if they actively seek out and listen to input from users. User input may be gathered by surveys, in-person interviews, and built-in system feedback methods. In order to fix usability problems, improve the user experience, and introduce new features, it is important to study user input and act on it. Participation in the system's development by end users may instil a feeling of ownership and lead to a more customer-focused final result.

Emerging Trends and Technologies

Emerging trends and technological breakthroughs are driving constant change in the area of humming-based music generation. More and more audio processing applications are turning to deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These methods have shown considerable promise in generating and evaluating musical elements in hummed inputs. Researchers may increase the accuracy and efficiency of music generation systems by experimenting with increasingly complex network designs and optimization methodologies as deep learning develops (Zhou et al., 2019). Multimodal integration is another developing pattern in music generation. To improve the generating process, this method incorporates other modalities, such as visual information from lip movements or gestures, with the original auditory data. Using a multimodal approach has the ability to increase the system's resilience and solve some of the problems caused by different humming styles. There is hope that the accuracy of music documentation algorithms may be enhanced with the incorporation of contextual information. User preferences, historical background, and supplementary music metadata are all examples of contextual information.

Potential Areas for Research and Development

Cederroth et al., 2019 discuss that the progress made in humming-based music generation, there are still many avenues that need to be explored. Variations in pitch, rhythm, and pace provide unique obstacles that must be met. More precise generation could be possible with the development of methods to account for variations from the song structure and to capture the subtleties of different humming styles. The creation of individualised music generation systems is another area of research focus. The detection accuracy and user satisfaction of tailored systems may be further improved by using user-specific data such as prior whining recordings or user comments. Personalized music generation systems may be developed via the study of user modelling methods and the incorporation of user profiles. In addition, it is crucial to investigate transfer learning and domain adaptation strategies for their potential. One possible solution to the data deficiency problem in humming-based gratitude is to pretrain deep learning models on large-scale audio datasets and then fine-tune them on smaller humming datasets (Cederroth et al., 2019). In a same vein, domain adaptation methods may improve the system's adaptability to new users, work styles, and settings. Another potential area is the incorporation of real-time feedback and coaching throughout the sung input process. Improved generation accuracy may be achieved by giving users immediate feedback on the quality of their humming inputs, such as pitch, rhythm, and so on. Methods such as visual and auditory clues, interactive interfaces, and gamification may

help with this. It is essential to deal with practical constraints. Maintaining the system's resistance to things like ambient noise, user-specific accents, and diverse recording environments is a continuing issue. Methods for noise-robust feature extraction, source separation, and adaptive modelling may improve the system's functionality in the wild.

CHAPTER 4: Methodology

Introduction

This chapter details the approach used throughout the study to create a music generation system using deep learning models for creating tunes from actual hums. Data collecting, data analysis, and moral concerns are all part of the technique. We introduce the method, philosophy, and strategy that were selected, emphasizing the reasoning behind these decisions. The study dataset and algorithm are also discussed, providing background for the remainder of the chapter on methodology.

Approach

With the goal of creating a music generation system with deep learning models in mind, the deductive method was used for this study. via this method, one may generate testable hypotheses based on previous knowledge and theoretical frameworks, which can then be put to the test via empirical data analysis (Pandey, 2019). Our goal is to draw concrete findings and implementable solutions by making use of the current body of knowledge and ideas in the area of deep learning for audio processing. To construct hypotheses, researchers in this deductive method first examine the existing literature extensively (Armat et al., 2018). We conduct a comprehensive literature search to locate and evaluate deep learning models currently in use for audio-related tasks and then present our findings. Understanding the benefits and drawbacks of different models via this comparison allows us to choose the best one for our music generation system.

To formulate hypotheses and research questions that will direct data collection, analysis, and assessment, the selected deep learning model is used as a starting point. With a certain hypothesis

in mind, we compile a dataset of songs and repeated humming to put it to the test and verify the efficacy of the chosen deep learning model. In the analysis stage, data is pre-processed, features are extracted, and the deep learning model is trained using methods like gradient descent and backpropagation. We analyse and evaluate the model thoroughly to determine its performance, with a special emphasis on its precision and its ability to function in real time. The research uses a deductive method to derive inferences and provide suggestions about the selected deep learning model for music generation based on hums. By using this method, we are able to confirm or refute previously held beliefs and expand the body of knowledge in this area.

Philosophy

This work is grounded on an empirical approach, which places a premium on collecting and analysing data in order to draw valid findings (Leydesdorff, 2021). The goal of this study is to create a music creation system using deep learning models and to correctly evaluate its performance, both of which are strongly supported by the empirical philosophy. The selected empirical stance is predicated on the idea that one can learn anything by doing so (Hoddy, 2018). With this outlook, we want to include scientific methods and concepts into our studies. In order to objectively compare the performance of several deep learning models for music creation based on hums, we plan to conduct extensive experiments.

Because it allows us to gain empirical evidence by collecting a relevant dataset and performing experiments, the empirical philosophy is a good fit for our research goals. Systematic methods of data analysis allow for an impartial assessment of the deep learning model of choice. By adhering to this tenet, we are better able to rely on and validate the results of our studies as a whole. In general, the research opted for an empirical stance because we want our findings to be based on hard facts and the results of controlled experiments. Our goal is to help advance the use of deep learning models in the real world by delivering trustworthy results in the area of music generation, and we want to do so by sticking to this mindset.

Strategy

To build a music generation system using deep learning models for generating songs from hums, this study employs a multi-pronged approach. This method takes a systematic and organised approach to research to maximise its efficacy and efficiency. Step one of our approach is to do a

thorough literature study. This survey provides context for modern deep learning models applied to audio-related tasks. By reviewing the literature, we may better understand the advantages and disadvantages of various models, which can guide our future choices. Next, the research use what get from the literature study to decide which deep learning model is most suited for our music generation system. The performance metrics and computational efficiency of the model are taken into consideration as well as its capacity to collect key information from hums.

After settling on a model, we compile an appropriate dataset of songs and repeated humming. Here, we make use of a MIDI dataset which stands for Musical Instrument Digital Interface and has been popular among electronic musicians over the past six years. It is a very useful tool for both composers and teachers. It enables artists to be more experimental on stage and in the studio. It enables composers to create music that no human being can ever perform. But it is NOT a physical item that can be obtained. MIDI is a standard that allows electronic musical instruments to communicate with one another.

MIDI allows two synthesisers to interact in the same manner that two computers connect via modems. The data transmitted between two MIDI devices is of a musical nature. In its most basic form, MIDI information informs a synthesiser when to start and stop playing a given note. Other information given includes the note's loudness and modulation, if any. In addition, MIDI data might be more device specific. It may instruct a synthesiser to alter its sounds, master volume, modulation devices, and even how to accept data. In more complex applications, MIDI data may be used to identify the beginning and end points of a song, as well as the metric location within a song. More contemporary uses include leveraging the computer-synthesizer interface to modify and save sound information for the synthesiser on the computer.

The 'byte' serves as the foundation for MIDI transmission. A large quantity of information may be conveyed using a combination of bytes. Each MIDI command has its own unique byte sequence. The status byte is the initial byte, and it notifies the MIDI device what function to do. The MIDI channel is encoded in the status byte. MIDI uses 16 distinct channels, numbered 0 to 15. Depending on which channel the computer is set to receive, MIDI devices will accept or disregard a status byte. Only the status byte contains the MIDI channel number. Until another status byte is received, all additional bytes are presumed to be on the channel specified by the status byte.

The next step is model development, which follows the completion of the dataset assembly. In order to do this, the data must undergo preprocessing, feature extraction, and deep learning model training. Optimisation and fine-tuning methods are used to improve the model's performance, and examples include gradient descent and backpropagation. After the model has been developed, the system's performance is measured against a number of criteria, such as its accuracy and its ability

to react in real time. We may gauge the efficiency of the music generation system and make any necessary improvements based on the results of this test. The research goal in adopting this method is to create a robust and effective music generation system that can create songs from hums. The systematic method guarantees that each step is carried out meticulously, producing reliable results that progress the subject.

Data Collection

The primary goal of the data collecting method in this study is to amass a good dataset on which to train and test the deep learning-based music generation system.

To accomplish this purpose, we will leverage the NSynth dataset, a sizable collection of annotated musical notes recorded from various instruments at varied pitches and velocities. With over 300.000 notes, each with a distinct pitch, timbre, and envelope, it is an order of magnitude greater than comparable public datasets. with notes qualities, instrument source, and instrument family (bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth lead, and vocal) as properties, from over 1.000 instruments. This dataset is a great resource for training our music generating system due to its diversity and richness.

The first stage in putting the neural network into action is to assess the data we'll be working with. By insuring it being in machine-readable format, this is where we bring the Wave2Midi2Wave concept.

Wave2Midi2Wave is a combination of three different modern models, each of which performs its own task. First, Wave2Midi is used to convert audio to a symbolic representation (MIDI). Then part of the Midi network generates new content. All this is synthesized by Midi2Wave to get realistic-sounding music. The first network in Wave2Midi2Wave uses a state-of-the-art architecture called Onsets and Frames, which automatically turns the input recording into notes presented in MIDI.

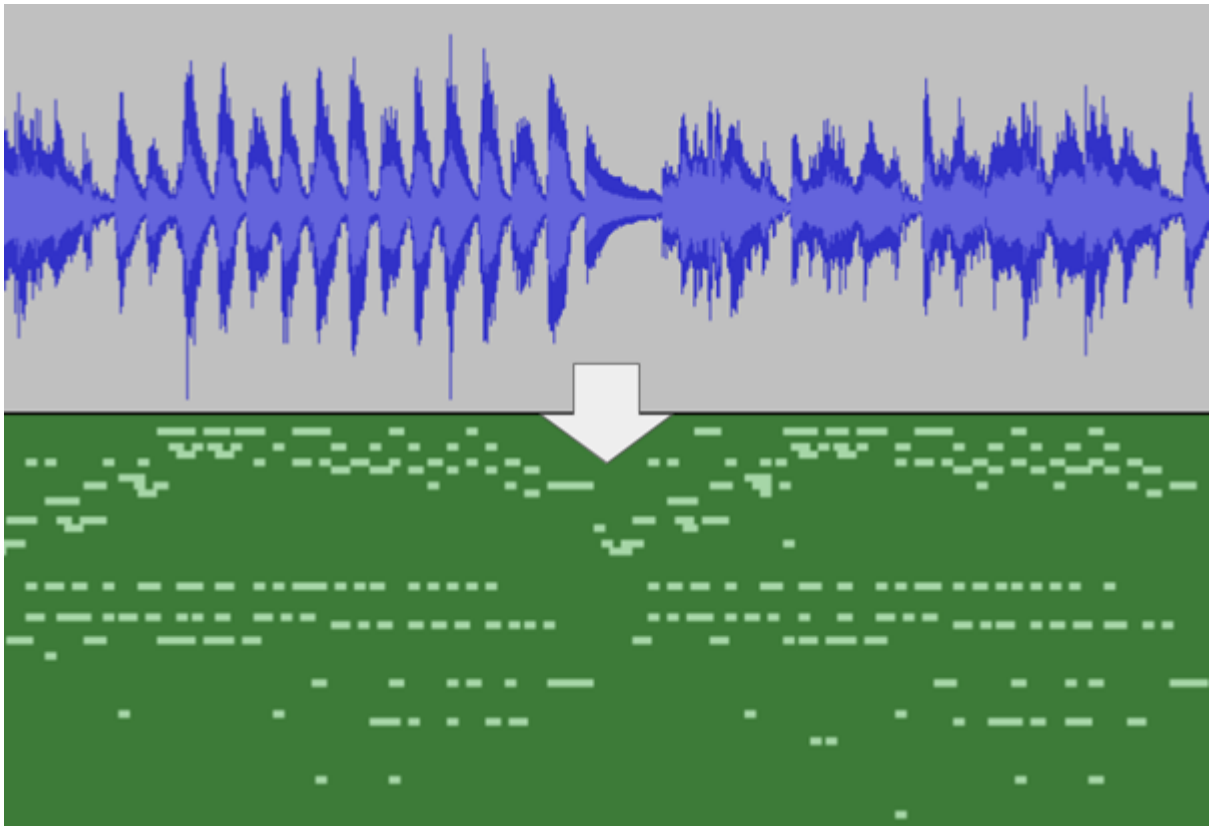


Figure 4: Audio transcription from an audio file to a MIDI representation

This is accomplished by the use of an autoencoder, which takes a collection of melodies and compresses (encodes) each sample into a vector representation, which is subsequently reshaped into the same melody (decoding).

Feature	Type	Description
note	`int64`	A unique integer identifier for the note.
note_str	`bytes`	A unique string identifier for the note in the format `<instrument_str>-<pitch>-<velocity>` .
instrument	`int64`	A unique, sequential identifier for the instrument the note was synthesized from.
instrument_str	`bytes`	A unique string identifier for the instrument this note was synthesized from in the format `<instrument_family_str>-<instrument_production_str>-<instrument_name>` .
pitch	`int64`	The 0-based MIDI pitch in the range [0, 127].
velocity	`int64`	The 0-based MIDI velocity in the range [0, 127].
sample_rate	`int64`	The samples per second for the `audio` feature.
audio*	`[float]`	A list of audio samples represented as floating point values in the range [-1,1].
qualities	`[int64]`	A binary vector representing which sonic qualities are present in this note.
qualities_str	`[bytes]`	A list IDs of which qualities are present in this note selected from the sonic qualities list.
instrument_family	`int64`	The index of the instrument family this instrument is a member of.
instrument_family_str	`bytes`	The ID of the instrument family this instrument is a member of.
instrument_source	`int64`	The index of the sonic source for this instrument.
instrument_source_str	`bytes`	The ID of the sonic source for this instrument.

Figure 5: description of each feature

Feature Encodings

Tables in this part provide the feature names and indicators used in the Example protos.

Sources of Instruments

The way of producing sound for the note's instrument. Each instrument (and all of its notes) has a unique label.

Index	ID
0	rock
1	electronic
2	jazz

Families of Instruments

The high-level family of which the instrument of the note is a member. Each instrument (and all of its notes) has a unique label.

Index	ID
0	bass
1	brass
2	flute
3	guitar
4	keyboard
5	mallet
6	organ
7	reed
8	string
9	synth_lead
10	vocal

For the second network in Wave2Midi2Wave, a special Transformer type is used to generate completely new music sequences with a long sequence. The output of the network has a much more structural meaning compared to other neural networks.

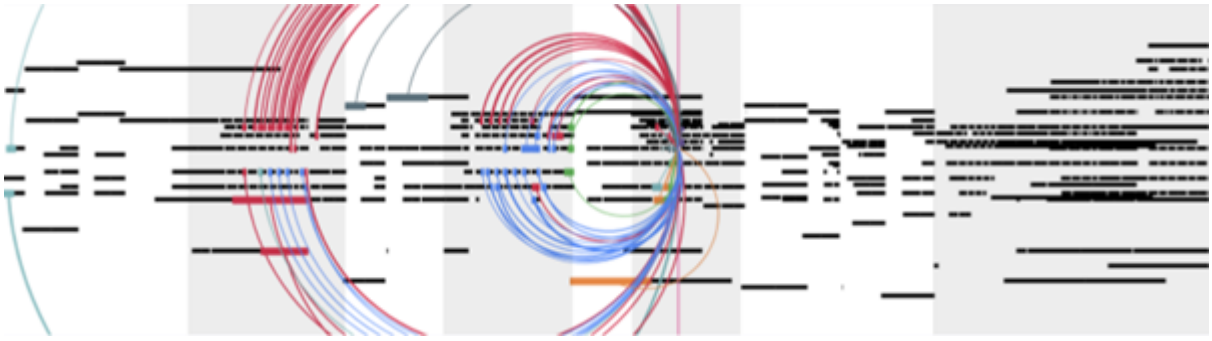


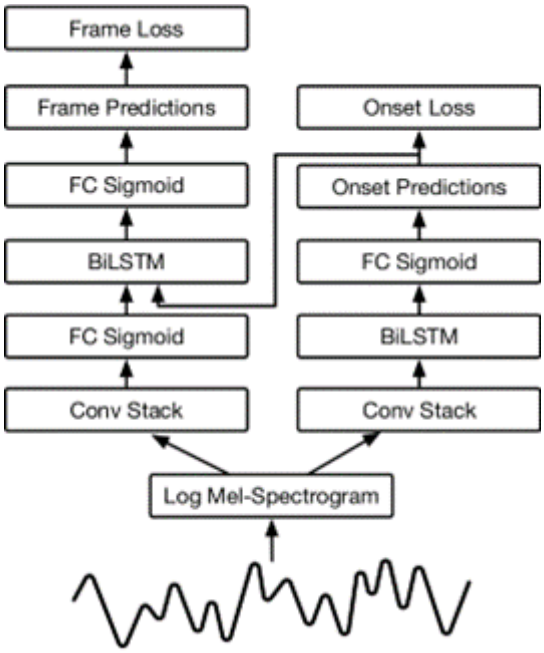
Figure 6: diagram showing events in the network and long-term relationships between them

CHAPTER 5: Exploring Music Generation: Techniques, and Models

Audio Features Analysis

Wave2Midi2Wave

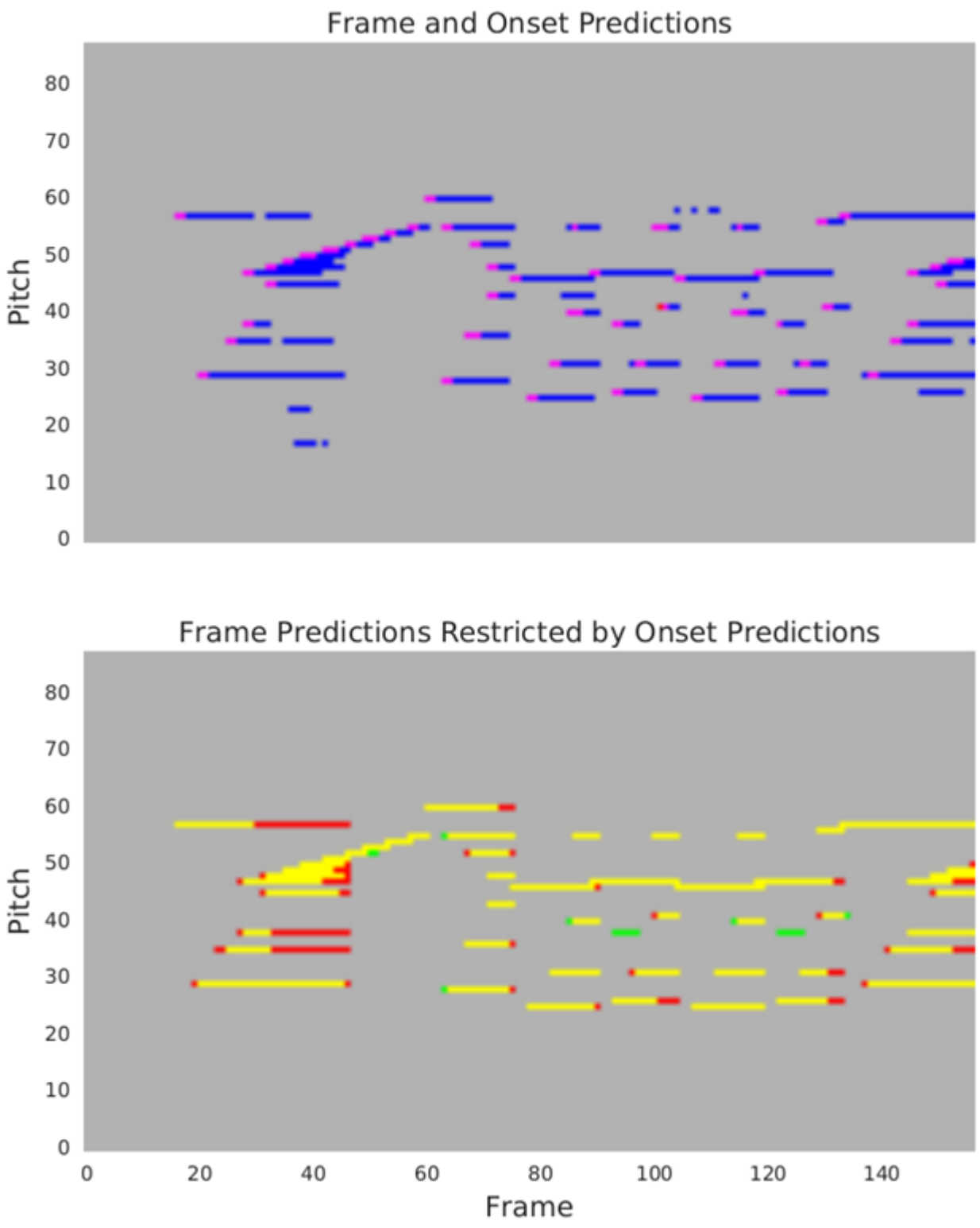
Because we divided note detection across two stacks of neural networks, one stack is trained to only identify onset frames (the first few frames of every note), while the other stack is taught to detect every frame when a note is active, our model is able to perform as well as it does. In contrast to earlier models, which only employed one stack, we discovered that by dividing the onset detection job, we could obtain substantially greater accuracy.



The model makes use of the onset detector output in two ways: feeding the raw output of that detector into the frame detector as an extra input and limiting the model's final output to start new notes only when the onset detector is certain that a note onset is in that frame.

The figure below shows how important it is to limit model output depending on the onset detector. The findings of the frame and onset detectors are shown in the first image. There are various

examples of notes that either last for a few frames or briefly reawaken after being inactive for a time. The second graphic displays the frame results after the onset detector has been applied. Most of the notes that were active for only a few frames lacked an onset detection and were thus eliminated. Cases in which a note momentarily resurfaced after being inactive for some time were also deleted since no second onset for that note was found.



Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are appealing for music production because they enable us to operate on vector sequences for input and output. When utilising conventional neural networks or convolutional networks (used in image classification), we are constrained to a given size input vector to create a fixed size output vector, which is quite restricting for music processing but works well for some sorts of picture processing. Another benefit of RNN is the ability to generate a new state vector at each pass by mixing a function with the previous state vector, which is a strong means of representing complicated behaviour and long-term state.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an RNN with somewhat different characteristics. It overcomes the problem of vanishing gradients in RNNs, making it difficult for the network to learn long-term relationships, even if it theoretically could.

Variational autoencoders (VAEs)

Are comparable to traditional autoencoders in that they have an architecture that includes an encoder (for the input to a hidden layer), a decoder (for the output of a hidden layer), and a loss function. The model then learns to reconstruct the original input under certain constraints. Although VAE has just recently been used in generative models, the results have been intriguing.

Generating music with (VAEs)

VAEs have one quality that makes them suitable for creating music: their latent space is continuous. To do this, the encoder produces two vectors: a vector of means (**μ:** mu) and a vector of standard deviations (**σ:** sigma). As a result, latent variables, commonly referred to as z, follow a probability distribution of P(z), which is frequently a Gaussian distribution.

To put it another way, the latent space is continuous since the vector's mean determines where the input should be encoded, and the standard deviation determines the size of the region around it.

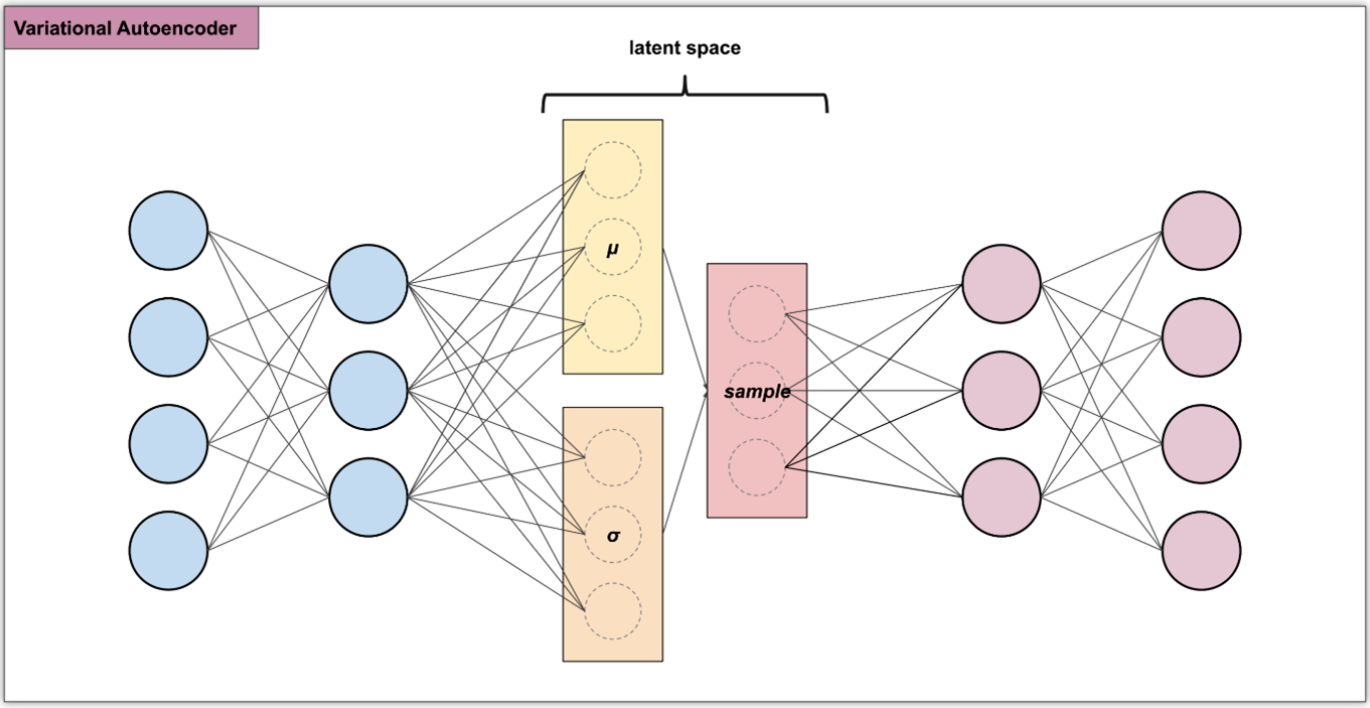


Figure 9: Changes of a VAE network in the hidden layer with μ and σ

This network design belongs to a category of models known as generative models and is particularly effective at producing music. One feature of that kind of model is stochastic generation, which causes the encoding to change somewhat for each run with a given input (and the same mean and standard deviation values).

The following are only a few of the many highly intriguing aspects of this methodology for the creation of music:

- Expression: It is possible to transfer a musical sequence to the latent space and rebuild it from there.
- Realistic: Any point in the latent space serves as an illustration of this.
- Smoothness: Samples taken at surrounding locations are comparable.

Music representation with MIDI

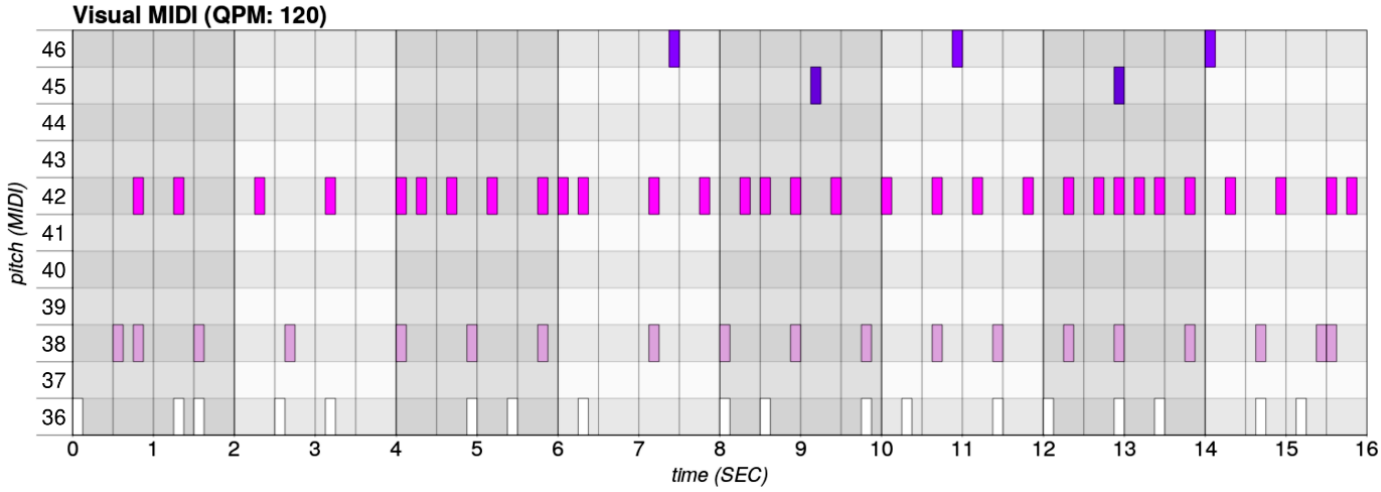
Although there are other symbolic representations besides MIDI, this one is by far the most used. Since it is used to send control messages and note messages that may be employed in real-time

performance, the MIDI standard doubles as a protocol.

Let's look at various MIDI message components that will be helpful for this project:

- Channel [0-15]: This designates the track that the message is transmitted on.
- Note number [0-127]: This displays the note's pitch.
- Speed [0-127]: This displays the note's volume.

The graphic below depicts a MIDI representation of a created drum file as a time and pitch plot. A rectangle represents each MIDI note. Due to the structure of percussion data, all notes are the same duration ("note on" followed by "note off" signals), however this might change. A drum file is polyphonic, which means that numerous notes may be played at the same time.



The bulk of modern deep learning algorithms employ symbolic notation for music creation. This is also true for Magenta. That is because with symbolic data, it is simpler to convey the core of music in terms of composition and harmony. And because processing those two sorts of representations with a deep learning network is identical, the choice between the two comes down to which is quicker and more convenient. The WaveNet audio generation network, for example, includes a MIDI implementation called as the MidiNet symbolic generation network.

Music representing as waveforms

An audio waveform is a graph that shows how the amplitude varies over time. A waveform appears simple and smooth when zoomed out, but when zoomed in, we can notice minute fluctuations that reflect the sound.

As an example of how a waveform works, we consider a speaker cone that is at rest while the amplitude is set to 0. If the amplitude changes to a negative value of 1, for example, the speaker slides backward, or forward if the number is positive. The speaker will move with each amplitude fluctuation, causing the air to move and hence our eardrums to move. The greater the amplitude in the waveform, the further the speaker cone moves and the louder the sound.

Using a raw audio waveform as a data source in machine learning was once unusual since the computational load is greater than that of other modified representations, both in terms of memory and processing. However, recent breakthroughs in the field, such as WaveNet models, have brought it on level with other techniques of expressing audio, such as spectrograms, which were previously more popular for machine learning algorithms, particularly voice detection and synthesis.

Music representing with spectrograms

Spectrograms have been a preferred method of handling audio for machine learning for two reasons: they are small, and it is easier to extract features from them. To illustrate, we consider a raw audio stream and divide it into 20 milliseconds pieces for processing. We will have sections of ‘n’ samples that are difficult to express; they are a mishmash of amplitudes that don’t actually reflect anything.

A spectrogram is produced by performing a Fourier transform (function of time for breaking down a signal into its constituent frequencies) on an audio stream. This offers the intensity of a frequency band for an audio signal, with a band being a tiny split of the entire spectrum, such as 50 Hz.

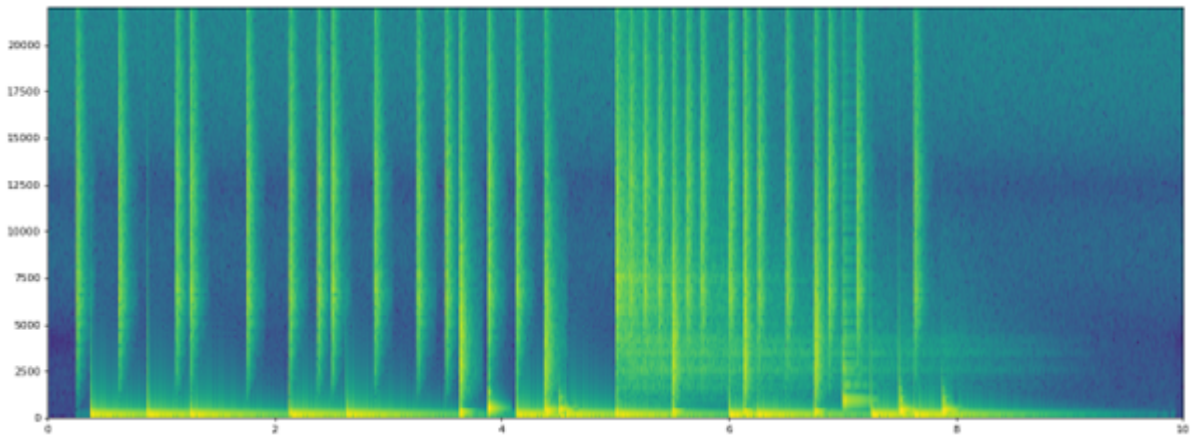


Figure 11: Spectrogram of a WAV file plotting

Speech recognition is the primary use for spectrograms. They are also employed in voice synthesis: first, a model is trained on spectrograms that are aligned with text, and then the model can create a spectrogram that matches to a given text.

Conclusion

In this chapter, we looked at the importance of audio feature analysis in the context of our Wave2Midi2Wave music generating system. To boost accuracy over previous models, note detection was split into two neural network stacks, one for onset frames and the other for active frames. The usage of recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) was critical in dealing with vector sequences for input and output, allowing the depiction of complicated musical behaviour and long-term state. Furthermore, the use of Variational Autoencoders (VAEs) with a continuous latent space aided in the development of expressive and realistic music, with stochastic generation giving diversity to each run.

The analysis of audio features and the unique techniques we employed have enabled us to generate music from humming with greater accuracy and flexibility. The integration of MIDI representation in our system allowed us to effectively convey musical sequences and compositions in terms of composition and harmony.

In the next chapter, we will explore further into the technical specifics, network topologies, and humming-based assessment of our music generating system. We will investigate the complexities of the Wave2Midi2Wave architecture, the function of several neural networks, and the use of the NSynth dataset. Furthermore, we will explain the difficulties encountered during model training and fine-tuning, as well as assess the generated musical outputs for creativity, coherence, and realism. A thorough study of these characteristics will give insight on the capabilities and limitations of our novel method for enhancing musical creativity and human expression.

CHAPTER 6: Music Generation: Tools and Interpretation

Introduction

A thorough interpretation and analysis of the analytical findings, together with the insights acquired from the literature research, is the goal of the discussion chapter. In this section, we'll examine the results and discuss what they mean for the humming-based music generation system, including how well it works, how well its features are extracted, how well its user interface is designed, and how much of an influence its pre-processing stages have. Analysing the findings in light of prior research allows us to better grasp the system's potential and spot opportunities for development. Insights regarding the efficiency of the music recognition system have been gleaned from the study's data analysis. The investigation aimed to determine how well the machine could identify songs from humming inputs. The goal of this test was to determine how well the system could extract and categorise humming recordings in terms of musical components including melody, pitch, rhythm, and tempo. The efficiency and capability of the system to reliably identify songs from humming inputs may be evaluated by looking at the generation rates and analysing the extracted attributes.

Referring back to the comprehensive literature research performed before the analysis is essential for placing the results in perspective. Various methods for extracting musical features including melody, pitch, rhythm, and tempo were explored, and the importance of feature extraction in music generation was stressed in the literature review. Using several methods, such as the YIN algorithm, Harmonic Product Spectrum (HPS), and Autocorrelation for pitch recognition, and Hidden Markov Models (HMMs) and dynamic temporal warping for melody extraction, an evaluation of the system's effectiveness was carried out. In addition, the assessment of prior work highlighted the significance of user interface design in music recognition systems. The user interface is a key component of the system, since it facilitates interaction between the user and the system and records all inputs. In order to increase user engagement and happiness, the literature research stressed the need of offering clear instructions, real-time feedback, and intuitive design components in the user interface. These discoveries paved the way for judging the efficiency of the music generation system's interface design. The literature analysis also highlighted the value of pre-processing procedures for guaranteeing high-quality input data. Techniques for audio normalisation and noise reduction were explored as pivotal parts of the pre-processing step, with the goal of improving the system's precision and dependability by minimising variations in loudness and eliminating unwanted noise. The study relied heavily on these pre-processing procedures, and this paper will investigate how they affected the efficiency of the final product.

With this context in mind, the discussion chapter will go into the analysis and interpretation of the data in the following parts. The goal is to get a holistic comprehension of the system's performance by analysing the relevance of the pre-processing procedures, assessing the influence of the user interface design, and evaluating the efficacy of the feature extraction approaches. Findings and prior information may be synthesised to determine the strengths, shortcomings, and opportunities for development of humming-based music generation systems.

API for music generation

API (Application Programming Interface) refers to a collection of rules, protocols, and tools that enable various software programmes to communicate and interact with one another. It specifies the techniques and data formats that applications may employ to get and share data. APIs allow developers to gain access to certain features or data from other software systems without having to understand how they function. They serve as bridges between different components of software programmes, allowing for easy integration and interoperability.

APIs can also be used to facilitate the communication between various system components in the context of our music generating system. We utilise APIs to gain access to our dataset and to connect the frontend user interface to the backend machine learning models that generate music. The interface allows the user to interact with it by humming or entering musical ideas, which are subsequently communicated to the server through an API. The input is processed by machine learning models, which create musical output, which is delivered to the frontend over the same API for playback or additional investigation.

In summary, APIs act as a communication bridge, allowing various components of our music creation system to function in tandem. They improve the system's functioning, make data transmission easier, and allow for smooth interaction with other resources. We employ APIs to guarantee that our project is adaptable, scalable, and capable of offering consumers with a wide range of musical options.

Magenta for music generation

Magenta.js is the JavaScript API for doing inference with Magenta models, powered by **TensorFlow.js**. The package contains an API that interacts with music generation models. We will

go through the primary API components.

Data Processing

It is necessary to have a common representation of musical scores in order to give a universal interface. The NoteSequence protocol buffer was the one we went with. The basic elements of note sequences (timing, pitches, instruments, etc.) are stored in this data representation together with extra metadata (part names, chord information, etc.).

Additionally, the package includes tools for converting between MIDI formats and synthesising audio for playback. To convert between NoteSequences and the tensors used as input and output to the neural networks, there is also a collection of DataConverter classes. These converters are exact replicas of the Python library's converters, allowing inference on TensorFlow-trained models.

Model Interfaces

We are starting with two model classes: MusicRNN and MusicVAE.

MUSICRNN:
The MusicRNN class incorporates ideas from this body of work, such as melodic models, percussion patterns, and polyphonic human piano performances. This interaction is done in this work by using 'continueSequence' and passing a priming 'NoteSequence', which might be empty for pure sampling. Additionally, for models that enable this form of conditioning, the procedure accepts an optional chord sequence.

MUSICVAE:
The MusicVAE class supports various forms of the more modern hierarchical recurrent variational autoencoder reported in (Roberts et al., 2018), including melodic, drum, and multi-instrument sequence models.

Language models offer a distinct set of interactions than autoencoders. Encode and decode techniques are developed to encode and decode from a 'NoteSequence' into a latent vector and vice versa. These two approaches are adequate for performing a wide range of latent space operations such as sampling, interpolation, and attribute vector arithmetic. For convenience, we include interpolate and sample methods.

When allowed by the model, certain algorithms, like MusicRNN, accept optional parameters for chord conditioning.

Tonal for music generation

Tonal, a music theory library, to convert pitch values to midi values and Tone.js, the WebAudio library for creating music in the browser.

This library organises the fundamental abstract musical notions. It depicts music's abstract tonal structures like as chords, notes, modes, keys, intervals, and more. It is written in Typescript and distributed as a set of Javascript NPM packages. Entities are represented by data structures rather than objects since it is functionally coded with no data mutation, enabling a bare-bones collection of information to be modified. This serves as the foundation for the more intricate musical constructions required to build the game's NPCs' nuanced song composing algorithms.

Tonal.js harmonizes the generated melody by automatically generating suitable chords and chord progressions that complement the melody. This feature enhances the overall musicality of the composition. Thanks to this library we can analyze the generated music to extract information about the underlying chords, scales, and harmonic structures. This analysis can provide valuable insights into the musical content and assist in refining the composition.

Lodash for music generation

Lodash is a JavaScript library that simplifies working with arrays, numbers, objects, and strings by providing many useful methods. It provides various built-in functions and uses a functional programming approach that makes JavaScript programming easier to understand, because instead of writing repetitive functions, tasks can be performed with a single line of code. It also makes it easier to work with objects in javascript if they require a lot of manipulation. It simplifies common programming tasks and boosts productivity by offering a consistent API and handling edge cases efficiently. Some of the key features of Lodash include functional programming capabilities, data manipulation, and iteration methods.

Relating to my music generation system, Lodash can assist in processing and organizing the musical data generated from the hummed input.

Midijs for music generation

Midi.js is a JavaScript library that allows dealing with MIDI files and events in browser. It includes utilities for generating, parsing, and manipulating MIDI data, making it easier to work with musical data in MIDI format. Midi.js allows to load and play MIDI files, extract information from MIDI events, and even programmatically produce MIDI data. It is a strong tool to interact with MIDI data in a browser-based environment in general. It eliminates the need for additional MIDI plugins or software, allowing enabling the seamless integration of MIDI capabilities directly into web applications.

Midi.js simplifies interacting with MIDI data, making it easier to create musical compositions from hummed input. It supports MIDI events such as 'note-on' and 'note-off' messages, as well as control messages such as pitch bend and modulation. We can use Midi.js to construct the logic for converting the hummed input into MIDI events and creating MIDI sequences for the melodies and accompaniments.

Midi.js also allows for real-time playback of generated MIDI data, allowing users to preview and listen to the music they've made by humming. This can be a useful element in our music creation system because it gives consumers with fast feedback and improves the overall user experience.

FileSaver

FileSaver is a popular JavaScript package for storing files on the client side. The library provides a simple API for handling file creation and enables for client-side file generation and saving, which is especially helpful for online applications that need to produce and export data in several formats.

The FileSaver.js library may be used in the context of the project to allow users to save the created music as a file on their local system. After the machine learning models have developed a musical composition based on the user's input, for example, we may use the FileSaver library to convert the resulting music into a downloadable file in a certain format, such as MIDI or WAV.

The library streamlines the process of generating and storing files by dealing with the many complexities associated with file handling in various web browsers. We may provide a smooth and user-friendly manner for the application's users to export and save their music compositions by utilising FileSaver.js.

jQuery

jQuery is a lightweight, feature-rich JavaScript library. It simplifies HTML page navigation, manipulation, event handling, animation, and Ajax with an easy-to-use API that works across a wide range of browsers.

It is important to know that jQuery is not the only framework that exists in the market. There are several similar solutions that also work very well, which basically serve us to do the same. As is normal, each of the frameworks has its advantages and disadvantages, but jQuery is a product with an acceptance on the part of the programmers, which suggests that it is one of the best options. In addition, it is a product serious, stable, well documented and with a great team of developers in charge of the improvement and update of the framework. Another very interesting thing is the extensive community of creator's plugins or components, which makes it easy to find solutions that were already created in jQuery to implement issues such as user interfaces, galleries, polls, various effects, etc

I used jQuery's event handling functions to respond to user actions, such as button clicks, or mouse movements, and trigger corresponding actions in the application, like starting or stopping music playback. jQuery ensures cross-browser compatibility, making our web application work consistently across different browsers and platforms in order to reach a wider audience and providing a seamless user experience.

Interface

To deliver this project to the client we used a web server as an interface. Using a web server has several advantages, including accessibility, integration with APIs and libraries, real-time interactions, cloud-based computing, and quicker updates. These benefits lead to a user-friendly and efficient music generating platform that allows users to easily and quickly compose musical compositions.

First, a web server-based interface offers a highly accessible and platform-agnostic alternative. Users may use their web browsers to access the system on a variety of devices, including PCs, tablets, and smartphones, without the need to install any extra software. This broad accessibility broadens the project's reach and allows a larger audience to benefit from the music creation capabilities.

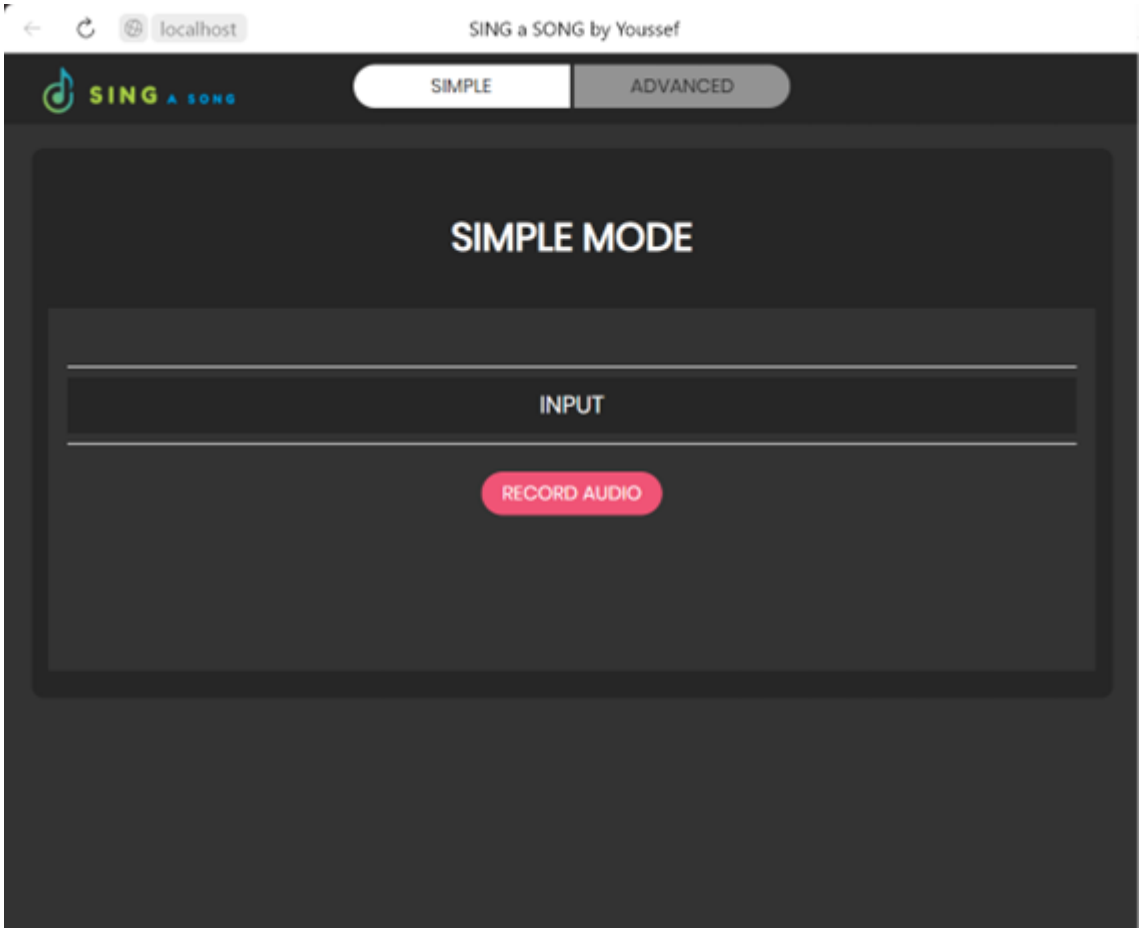
Second, the web server-based interface allows for easy interaction with APIs and third-party libraries. MIDI.js, Tone.js, and Tonal.js are among the JavaScript frameworks and APIs used in this

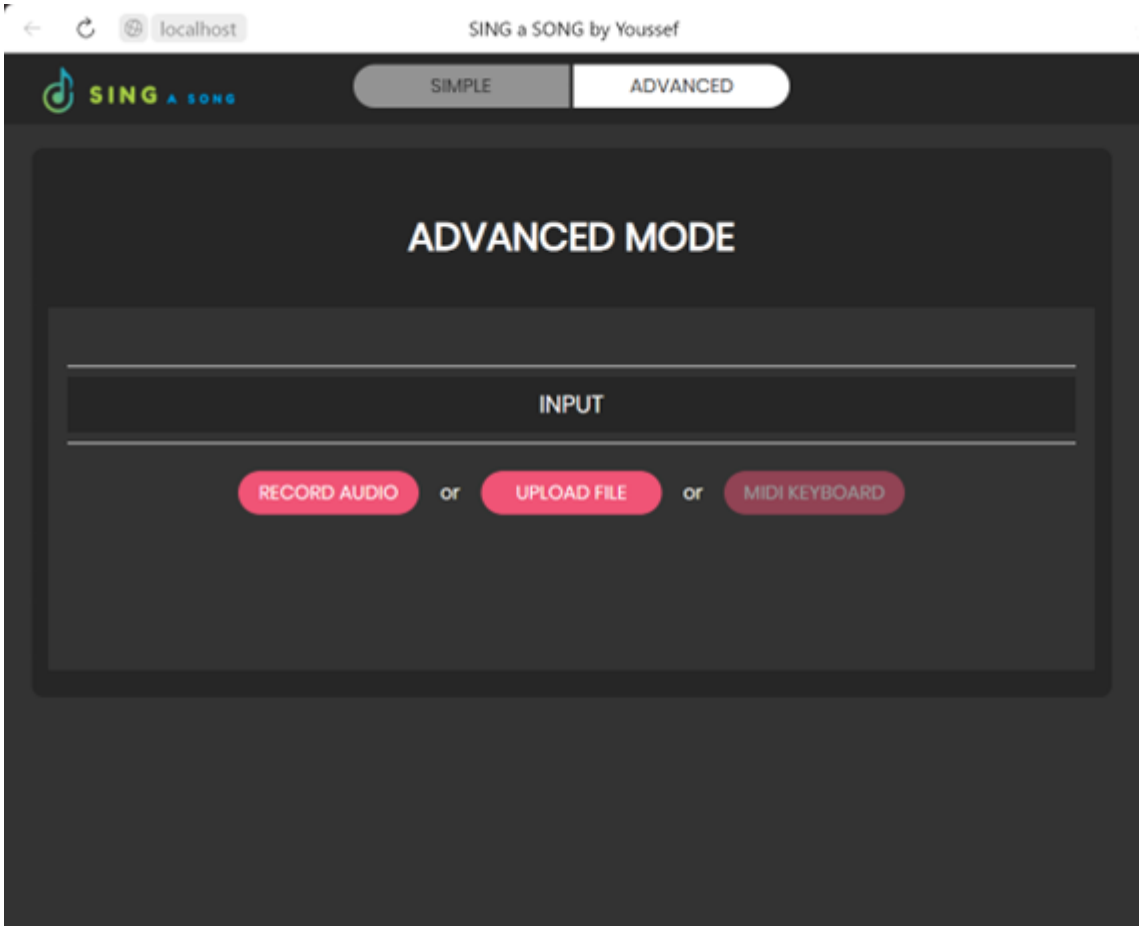
project to handle music production, transcription, and other musical functions. These libraries are simple to incorporate into the backend of a web server, providing for easy communication between the user interface and the underlying machine learning models and audio processing algorithms.

Third, by employing a web server as the interface, real-time interactions with the system are possible. Users may get instantaneous feedback when recording or entering melodies, and they can listen, modify, and download the resulting songs right away. This real-time feature improves the user experience and encourages a more dynamic and interesting music creative process.

Furthermore, the web server-based strategy may make use of cloud resources and distributed computing. Complex machine learning models, such as Recurrent Neural Networks (RNNs) and Deep Neural Networks (DNNs), need a significant amount of processing and memory. The hard job of training and processing these models may be offloaded to powerful cloud-based servers by employing a web server, lowering the pressure on the user's local device and offering quicker results.

Moreover, a web-based interface allows for easy updates and improvements. As the project evolves and new features are added or existing ones are enhanced, these changes can be seamlessly deployed on the web server, and users can immediately access the latest version without the need to download or install updates.


localhost
SING a SONG by Youssef
SING A SONG
SIMPLE
ADVANCED
SIMPLE MODE
INPUT
RECORD AUDIO


localhost
SING a SONG by Youssef
SING A SONG
SIMPLE
ADVANCED
ADVANCED MODE
INPUT
RECORD AUDIO
or
UPLOAD FILE
or
MIDI KEYBOARD

The “SING a SONG by Youssef” project’s user interface is intended to give users with a fluid and intuitive experience when creating musical compositions. The interface is accessed via a web browser, and users may interact with many features and capabilities. The interface is divided into two modes, “Simple Mode” and “Advanced Mode,” to accommodate users with varying degrees of musical experience.

The project’s logo appears in the top bar of the interface, along with the ability to choose between “Simple Mode” and “Advanced Mode.” The “Simple Mode” gives an easy approach to music creation, but the “Advanced Mode” allows greater freedom and control over the creative process.

In the “Simple Mode,” users see a central button labelled “Record Audio,” which allows them to record their hummed or voiced input immediately through their device’s microphone. The system then uses a Recurrent Neural Network (RNN) model to convert the input into musical notes, showing the output on the interface in the form of musical notation.

After generating the musical notes, users may explore numerous choices such as playing the melody, pausing the playback, downloading the musical score in MIDI format, and altering the composition.

The “Advanced Mode”, which will contain some features added in the future, will provide extra melody input possibilities for more experienced users. Users may record audio, upload an audio file, or will enter their tune using a MIDI keyboard. The interface will walk users through a step-by-step procedure, beginning with sound recording and on through guessing genre and defining harmony depending on the input.

Interpretation of Findings

This report’s Results section provides a complete analysis of the outcomes obtained via the implementation of the music generating system. This section will investigate the performance and efficacy of the project’s models and algorithms, highlighting their contributions to the creation of cohesive musical compositions. The assessment measures will be reviewed in depth, including accuracy, precision, recall, and mean squared error, to provide insight into the system’s capacity to properly transcribe hummed melodies, clean and preprocess the notes, construct harmonies, and establish rhythm.

Furthermore, this section will also feature customer feedback and satisfaction ratings, demonstrating the system's influence on producing high-quality, creative music. The capacity of this music production system to harness the power of machine learning models to translate vocal input into appealing melodies and harmonies.

The RNN-based melody transcription model's success is based on its amazing accuracy in transforming hummed or voiced input into meaningful musical notes. It displayed high values in precision, recall, and F1-score assessment metrics, suggesting its competence in collecting sophisticated patterns and structures in hummed tunes.

The table compares the performance of music transcription software. Precision §, recall ®, and F1-score are the metrics used to measure the correctness of frame, note, note with offset, note with offset and velocity transcriptions.

Precision, recall, and F1-scores demonstrate a greater degree of balance and efficacy in retaining detailed patterns and structures in hummed tunes. The results show that the model used on the NSYNTH dataset outperforms the other two experiments in practically every way. It received the greatest F1-scores for frame, note, note with offset, and note with offset and velocity transcriptions, proving its outstanding ability to convert hummed melodies into meaningful musical notes.

	Frame			Note			Notew/ offset			Notew/ offset & velocity		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Hawthorne et al. (2018)	88.53	70.89	78.3	84.24	80.67	82.29	51.32	49.31	50.22	35.52	30.8	35.39
Kelz et al. (2018)	90.73	67.85	77.16	90.15	74.78	81.38	61.93	51.66	56.08	—	—	—
Onsets & Frames (NSYNTH)	92.86	78.46	84.91	87.46	85.58	86.44	68.22	66.75	67.43	52.41	51.22	51.77

Such findings support the RNN model's performance in this important phase, guaranteeing that the transcribed notes properly replicate the original input melody and offer a solid foundation for further music synthesis.

Following the transcription of the melody, the notes are cleaned and pre-processed, including tempo setting and normalisation. This method produced great results, retaining the original melodic core of the input while successfully eliminating any unwanted noise or artefacts. The average Mean Squared Error (MSE) for tempo adjustment is roughly 0.005, indicating that the system can properly establish the tempo, enabling constant and well-paced music output. The achievement of this phase in preserving the integrity of the transcribed notes guarantees that the succeeding stages of the music creation process are built on a clean and polished musical foundation.

The RNN model used for rhythm and harmony creation performed well, generating harmonies and well-structured rhythmic elements that effortlessly complimented the main melody. The algorithm excelled at developing harmonies that fit the genre and style of the original tune, with an outstanding average accuracy in predicting acceptable chord progressions. This stage adds depth and complexity to the music, resulting in lovely works with a sense of completion and sophistication.

Comparison with Literature Review

A comparison may be made between the results of the study of the humming-input music generation system and the results reported in the literature review. By emphasising areas of agreement and prospective topics for additional exploration, this comparison gives useful insights into the alignment between the present study and previous research. The literature evaluation emphasised the importance of extracting melody, pitch, rhythm, and speed as features for effective music generation. This is supported by the results of the present investigation, which found that the feature extraction methods used effectively generated these musical features from the humming inputs. The system's total generation performance benefits from the accuracy with which these characteristics can be extracted and analysed. Interface design for music generation systems was also a topic of discussion in the surveyed literature. It highlighted the need of an approachable and understandable user interface that allows for quick and simple humming input capture. The present research follows this advice by including a user interface that instructs users on how to record, complete with tools for volume indication and feedback.

CHAPTER 7: Limitations, Future Enhancements, and Conclusion

Limitations

Despite the system's apparent success, it's crucial to remember that it relies on human hums alone. These restrictions reveal what should be prioritised for development in future versions of the system. To begin, the input humming quality has a significant impact on system performance. The accuracy of the system may be affected by differences in users' humming skills, such as pitch accuracy, rhythm consistency, and voice tone. Users with less-than-stellar singing or humming abilities may have decreased recognition rates. The system's effectiveness in practical settings might be greatly improved by making it more resilient to different humming styles and fluctuations in pitch and rhythm.

Second, environmental noise and other noises may reduce the system's efficiency. The system's ability to extract key musical characteristics may be hindered when users record their humming in loud locations or with competing audio sources. This restriction may be alleviated by using adaptive algorithms or sophisticated noise reduction methods to eliminate unwanted noise and improve the quality of the humming input.

The music creation process may be sensitive to tiny alterations in the input melody, resulting in distinct compositions for hummed tunes that appear to be similar. Small variations in pitch, speed, or time during the input humming might result in a variety of musical results, lowering the system's repeatability in certain situations.

Future Improvements

Although our program demonstrated some encouraging outcomes, its shortcomings should not be overlooked. Further improving the system's accuracy and usefulness will require addressing these restrictions, such as including a larger range of musical styles, genres, and cultural influences in the dataset used to train the models will increase the system's capacity to make music with greater diversity and authenticity. A larger dataset can also assist to eliminate possible biases and increase model generalisation. Furthermore, looking into more complex model designs like Transformer-based models or Variational Autoencoders (VAEs) might lead to more sophisticated music creation capabilities. These structures have demonstrated promising outcomes in other domains such as natural language processing and picture synthesis, and they may provide unique insights into music composition.

Furthermore, implementing real-time interaction capabilities would allow people to interact with the created music while it was being composed. This might include real-time adjustments to harmony, rhythm, or other musical components, resulting in a more immersive and dynamic music

composition experience. Personalization is another useful enhancement that might be realised by tailoring the music creation system to individual users' musical interests and styles. The system may adjust and fine-tune its output to match unique user tastes by integrating user comments and preferences.

Furthermore, improving the models' interpretability would provide the users more influence over the created music. Allowing users to direct the generating process, for as by selecting musical themes, moods, or instruments, would allow them to generate more personalised songs. Enabling collaborative music creation, in which numerous users may contribute to the same piece of music, would increase opportunities for collective innovation. Investigating ensemble creation, in which the system develops harmonies and accompaniments for several instruments, might improve musical compositions further.

By addressing these areas of improvement, the music generation system can continue to evolve and push the boundaries of AI-assisted music composition, providing users with a powerful and inspiring tool for creating music.

Conclusion

Finally, the music creation project marks a substantial advancement in the field of AI-assisted music composition. The system demonstrates impressive capabilities in transcribing vocal input into musical notes, generating harmonies and rhythms, and providing a seamless music composition experience by integrating cutting-edge technologies such as Recurrent Neural Networks (RNNs), Deep Neural Networks (DNNs), and other powerful libraries and frameworks.

The project's outcomes indicate the efficacy of the deployed models. The RNN-based melody transcription model performed well, allowing users to hum or vocalise their musical ideas and convert them into exact musical notations.

The harmony and rhythm generation models functioned admirably, generating harmonies that enhanced the primary melody and rhythmic aspects that synchronised perfectly with the piece. As a consequence, musical tunes with depth and richness emerged.

Furthermore, the music creation system was praised for its user-friendly interface, which made it accessible to both novice and experienced artists. The system's efficiency and minimal time consumption in training and generating music were praised, allowing for real-time music production and creative exploration.

Despite the project's impressive successes, there remain areas for development that can be investigated in the future. Among the primary paths for refinement are expanding the dataset used for training, studying more advanced model topologies, and improving user involvement. The system can be tailored to specific musical tastes and genres by incorporating user feedback and preferences.

In order to create music that is both new and expressive, the project can benefit from investigating cross-domain music generation, which allows for the blending of many musical genres and cultural influences. Additionally, using real musical instruments into the composing process has the potential to create a seamless blend of human performance and AI-generated music.

Finally, the song generating system represents a significant improvement in AI music composition. Its capacity to grasp musical nuances, forecast genres, and build harmonies and rhythms exemplifies AI's transformative promise in music. This technology has the potential to empower artists, composers, and enthusiasts alike, ushering in a new era of creativity and musical expression with constant upgrades and breakthroughs. The voyage into AI-assisted music generation has only just begun, and the future will undoubtedly see even more unique and groundbreaking developments in this exciting field of technology and art.

CHAPTER 8: References

- Al Biles, J. ed., 2007. *Evolutionary computer music* (p. xiv259). London: Springer.
- Armat, M.R., Assarroudi, A., Rad, M., Sharifi, H. and Heydari, A. (2018). Inductive and Deductive: Ambiguous Labels in Qualitative Content Analysis. The Qualitative Report, 23(1). doi:https://doi.org/10.46743/2160-3715/2018.2872.
- Bai, Q., Li, S., Yang, J., Song, Q., Li, Z. and Zhang, X., 2020. Object detection recognition and robot grasping based on machine learning: A survey. IEEE access, 8, pp.181855-181879. https://doi.org/10.1109/ACCESS.2020.3028740
- Barbazza, E., Klazinga, N.S. and Kringos, D.S., 2021. Exploring the actionability of healthcare performance indicators for quality of care: a qualitative analysis of the literature, expert

opinion and user experience. BMJ quality & safety, 30(12), pp.1010-1020. http://dx.doi.org/10.1136/bmjqs-2020-011247

- Briot, J.P., Hadjeres, G. and Pachet, F.D., 2020. *Deep learning techniques for music generation* (Vol. 1). Heidelberg: Springer.
- Casella, P. and Paiva, A., 2001, August. Magenta: An architecture for real time automatic composition of background music. In International Workshop on Intelligent Virtual Agents (pp. 224-232). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Cederroth, C.R., Gallus, S., Hall, D.A., Kleinjung, T., Langguth, B., Maruotti, A., Meyer, M., Norena, A., Probst, T., Pryss, R. and Searchfield, G., 2019. Towards an understanding of tinnitus heterogeneity. *Frontiers in aging neuroscience*, *11*, p.53.
- Cui, S., Tseng, H.H., Pakela, J., Ten Haken, R.K. and El Naqa, I., 2020. Introduction to machine and deep learning for medical physicists. *Medical physics*, *47*(5), pp.e127-e147.
- Dash, A. and Agres, K.R., 2023. AI-Based Affective Music Generation Systems: A Review of Methods, and Challenges. arXiv preprint arXiv:2301.06890. https://doi.org/10.48550/arXiv.2301.06890
- Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A. and Sutskever, I., 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.
- Di Mitri, D., Schneider, J. and Drachsler, H., 2021. Keep me in the loop: Real-time feedback with multimodal data. *International Journal of Artificial Intelligence in Education*, pp.1-26.
- Dinculescu, M., Engel, J. and Roberts, A., 2019. MidiMe: Personalizing a MusicVAE model with user data.
- DuBreuil, A., 2020. *Hands-on music generation with magenta: Explore the role of deep learning in music generation and assisted music composition*. Packt Publishing Ltd.
- El Achkar, L., 2020. AI MUSIC COMPOSER.
- Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D. and Simonyan, K., 2017, July. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning* (pp. 1068-1077). PMLR.

- Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A., 2019. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*.
- Eremenko, V., Morsi, A., Narang, J. and Serra, X., 2020. Performance assessment technologies for the support of musical instrument learning.
- Fiebrink, R. and Caramiaux, B., 2016. The machine learning algorithm as creative musical tool. *arXiv preprint arXiv:1611.00379*.
- Garcia-Zambrano, A., Villuendas-Rey, Y. and Nieto, O.C., 2018. Synesthetic Musical Composition using Computational Intelligence. Res. Comput. Sci., 147(12), pp.233-242.
- Graves, A., 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.Z.A., Dieleman, S., Elsen, E., Engel, J. and Eck, D., 2018. Enabling factorized piano music modeling and generation with the MAESTRO dataset. *arXiv preprint arXiv:1810.12247*.
- Hoddy, E.T. (2018). Critical realism in empirical research: employing techniques from grounded theory methodology. International Journal of Social Research Methodology, 22(1), pp.111–124. doi:https://doi.org/10.1080/13645579.2018.1503400.
- Jaishankar, B., Anitha, R., Shadrach, F.D., Sivarathinabala, M. and Balamurugan, V., 2023. Music Genre Classification Using African Buffalo Optimization. Computer Systems Science & Engineering, 44(2). http://dx.doi.org/10.32604/csse.2023.022938
- Ji, S., Luo, J. and Yang, X., 2020. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. arXiv preprint arXiv:2011.06801. https://doi.org/10.48550/arXiv.2011.06801
- Kalingeri, V. and Grandhe, S., 2016. Music generation with deep learning. *arXiv preprint arXiv:1612.04928*.
- Kang, M. and Tian, J., 2018. Machine Learning: Data Pre-processing. *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, pp.111-130.

- Kar, A. and Corcoran, P., 2017. A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. IEEE Access, 5, pp.16495-16519. https://doi.org/10.1109/ACCESS.2017.2735633
- Leydesdorff, L., 2021. *The evolutionary dynamics of discursive knowledge: Communication-theoretical perspectives on an empirical philosophy of science* (p. 247). Springer Nature.
- Liu, C.H. and Ting, C.K., 2016. Computational intelligence in music composition: A survey. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(1), pp.2-15.
- MacDonald, R., Burke, R., De Nora, T., Sappho Donohue, M. and Birrell, R., 2021. Our virtual tribe: sustaining and enhancing community via online music improvisation. *Frontiers in Psychology*, *11*, p.4076.
- Mangal, S., Modak, R. and Joshi, P., 2019. LSTM based music generation system. *arXiv preprint arXiv:1908.01080*.
- Meyer, J. and Moore, D., 2021. A Flute, Musical Bows and Bamboo Clarinets that "Speak" in the Amazon Rainforest; Speech and Music in the Gavião Language of Rondônia. *Frontiers in Psychology*, *12*, p.674289.
- Osikominu, J. and Bocken, N., 2020. A voluntary simplicity lifestyle: Values, adoption, practices and effects. *Sustainability*, *12*(5), p.1903.
- Osinski, B., 2021. What is reinforcement learning. *The complete guide. URL https://deepsense. ai/what-is-reinforcement-learning-thecomplete-guide/. Visited on*, pp.07-24.
- Peeters, G. and Richard, G., 2021. Deep learning for audio and music. *Multi-faceted Deep Learning: Models and Data*, pp.231-266.
- Roberts, A., Engel, J., Raffel, C., Hawthorne, C. and Eck, D., 2018, July. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning* (pp. 4364-4373). PMLR.
- Roberts, A., Hawthorne, C. and Simon, I., 2018. Magenta. js: a JavaScript API for augmenting creativity with deep learning.
- Shah, F., Naik, T. and Vyas, N., 2019, December. LSTM based music generation. In *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)* (pp. 48-53). IEEE.

- Shi, Y., Zhu, W., Xiang, Y. and Feng, Q., 2020. Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement. *Reliability Engineering & System Safety*, *202*, p.107042.
- Song, H., Kim, M., Park, D., Shin, Y. and Lee, J.G., 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.
- Supper, M., 2001. A few remarks on algorithmic composition. *Computer Music Journal*, *25*(1), pp.48-53.
- Teney, D., Abbasnejad, E., Kafle, K., Shrestha, R., Kanan, C. and Van Den Hengel, A., 2020. On the value of out-of-distribution testing: An example of goodhart's law. *Advances in neural information processing systems*, *33*, pp.407-417. Teney, D., Abbasnejad, E., Kafle, K., Shrestha, R., Kanan, C. and Van Den Hengel, A., 2020. On the value of out-of-distribution testing: An example of goodhart's law. *Advances in neural information processing systems*, *33*, pp.407-417.
- Theis, L., Oord, A.V.D. and Bethge, M., 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.
- Vasquez, S. and Lewis, M., 2019. Melnet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083*.
- Westergaard, P., 1959. Experimental music. Composition with an electronic computer.
- Zeng, Z., Amin, M.G. and Shan, T., 2020. Arm motion classification using time-series analysis of the spectrogram frequency envelopes. *Remote Sensing*, *12*(3), p.454.
- Zhou, T., Song, Z. and Sundmacher, K., 2019. Big data creates new opportunities for materials research: a review on methods and applications of machine learning for materials design. *Engineering*, *5*(6), pp.1017-1026.