IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A FRAMEWORK FOR PIANO NOTES ANALYSIS USING DIGITAL SIGNAL PROCESSING

Leo Prasanth L.[1*], Uma E.[2]

[1]Research Scholar, [2]Assistant Professor

[1]Department of Information Science &Technology,

[1]College of Engineering, Anna University, Chennai, Tamilnadu, India

Abstract

Music is not just an art, music is an expression of the human condition. When an artist is making a song you can often hear the emotions, experiences, and energy they have in that moment. Music connects people all over the world and is shared across cultures. Music is a ubiquitous and vital part of the lives of billions of people worldwide. Musical creations and performances are among the most complex and intricate of our cultural artifacts and the emotional power of music can touch us in surprising and profound ways. This project takes song as an input, extracts the features and detects and identifies the notes, each with a duration using Digital Signal Processing (DSP). DSP take real-world signals like voice, audio, video, temperature, pressure or position that have been digitized and then mathematically manipulate them. A DSP is designed for performing mathematical functions like ”add”, ”subtract”, ”multiply” and ”divide” very quickly. Signals need to be processed so that the information that they contain can be displayed, analyzed or converted to another type of signal that may be of use. The experiment is done with piano songs with unknown notes with the proposed algorithm. Therefore, this project might be of great use for those who consider music as their passion and have mere knowledge about music.

Keywords: DIGITAL SIGNAL PROCESSING (DSP), PIANO NOTES, FOURIER TRANSFORM.

1. INTRODUCTION

Digital Signal Processing (DSP) is the process of analyzing and modifying a signal to optimize or improve its efficiency or performance. It involves applying various mathematical and computational algorithms to analog and digital signals to produce a signal that’s of higher quality than the original signal. DSP is primarily used to detect errors, and to filter and compress analog signals in transit. It is a type of signal processing performed through a digital signal processor or a similarly capable device that can execute DSP specific processing algorithms. Typically, DSP first converts an analog signal into a digital signal and then applies signal processing techniques and algorithms. For example, when performed on audio signals, DSP helps reduce noise and distortion. Some of the applications of DSP include audio signal processing, digital image processing, speech recognition, biomedicine and more.

1.1. FOURIER TRANSFORM

A Fourier transform provides the means to break up a complicated signal, like a musical tone, into its constituent sinusoids. This method involves many integrals and a continuous signal. We want to perform a Fourier transform on a sampled (rather than continuous) signal, so we have to use the Discrete Fourier Transform instead. The most common implementation of the DFT is the Fast Fourier Transform. The FFT arrives at the same result as the DFT, but the DFT has a run time $O(N^2)$.

While the FFT has a runtime of O(NlogN). When talking about the FFT, it's important to clearly define the terms used. Specifically, the input to an FFT has a number of parameters:

fs: sampling frequency. This number is how many samples per second were taken when converting the analog signal to a digital one. It is measured in Hz. Sampling frequency is sometimes also written as fs.

n: the number of data points in the input.

T: the total time we're sampling over. This is measured in seconds. It can be calculated by dividing n by fs.

f: frequency resolution. This is defined as 1/T.

Our goal is to find the frequencies of the constituent sinusoids of the musical tone, because this relates to how its pitch is perceived. For the FFT to be useful, we have to have the FFT operate on a long enough time (T) so that it can distinguish between an instrument's lower pitches. The frequency resolution is the inverse of T, due to the fact that in order to resolve a frequency accurately, enough time has to pass to complete one full cycle at that frequency.

1.2. NEED FOR THE SYSTEM

The revolution in music distribution and storage brought about by personal digital technology has simultaneously fueled tremendous interest in and attention to the ways that information technology can be applied to this kind of content. This project can be treated as a box, where you give any song as input and get the features of the song as output. The aim of this project is to propose methods to analyze and describe a signal, from where the musical parameters can be easily and objectively obtained, in a sensible manner.

1.3. PROBLEM STATEMENT

Songs play a vital role in our day to day life. A song contains basically two things, vocal and background music. Where the characteristics of the voice depend on the singer and in case of background music, it involves mixture of different musical instruments like piano, guitar, drum, etc. To extract the characteristic of a song becomes more important for various objectives like learning, teaching, composing. Hence this project attempts to derive the notes from the signal so that a musical score could be produced.

1.4. OBJECTIVES

- The main objective of this project is to create an aid tool for learning for Musicians, Producers, Composers, DJs, Remixer, Teachers and Music Students.
- The software system accepts the signal as a digitized waveform representing an acoustic music signal and attempts to extract the notes from it in order to generate a musical score.
- The signal processing algorithm includes event detection, or precisely where within the signal the various notes actually begin and end, and pitch extraction or the identification of the pitches being played in each interval.
- The event detection is carried out using the time domain analysis of the signal and the frequency identification as follows.

2. LITERATURE SURVEY

The system proposed by Ivan P. Yamshchikov et al., [12] described about a new artificial neural network architecture that helps generating longer melodic patterns is introduced alongside with methods for post-generation filtering. A recurrent highway gated network combined with a variation auto encoder is used in the proposed variation auto encoder supported by history technique. The use of this architecture in conjunction with filtering heuristics results in pseudo-live, acoustically attractive, melodically diverse music. Because the issue configuration for note-by-note music creation and the problem setup for word-by-word text generation are similar, it's worth revisiting some of the strategies that have proven successful in generative natural language processing jobs. A variation auto encoder is used in this system (VAE). A VAE uses a variation technique for latent representation learning and makes assumptions about the distribution of latent variables. This results in an additional loss component and the Stochastic Gradient Variation Bayes training procedure (SGVB).As a result, a generative VAE produces instances that are comparable to those obtained from the input data distribution. It also provides significant control over the created output's parameters. This potentially allows for controlled music output, making the idea of using a VAE-based approach to generate music particularly appealing.

A proprietary dataset of four gigabytes of midi files containing songs from various eras and genres was employed in the studies. The data was already there, but it required extensive treatment. A single midi file can have numerous tracks with useful information as well as tracks of negligible value. As a result, the files were divided into distinct tracks. Because data normalization is frequently required to aid learning, the following normalization methods were conducted to each track separately. Each note in a midi file is specified by multiple factors, including pitch, length, and strength, as well as track (e.g., the instrument that is playing the note) and file parameters. Despite the importance of nuancing in musical compositions, the power of the notes was ignored in our experiments. The melodic patterns determined by the pitches and the temporal dimensions of the notes and pauses in between are the subject of this article. Every track's median pitch was shifted to the fourth octave. Finally, tracks with extremely low entropy were removed from the training data to avoid the model from over-fitting and to make the input diverse enough. Because tracks are created note by note, having a large number of tracks with low pitch entropy would substantially reduce the output quality. The final dataset, which included over 15,000 normalized tracks, was utilized to teach more people. For each note in each track, a concatenated note embedding was created. This embedding includes the note's pitch, octave, and a delay corresponding to the note's length. Each individual MIDI track had meta-information contained in it as well.

2.1. A GENERATIVE MODEL FOR RAW AUDIO USING WAVENET

The system proposed by Verma et al., [10] described WaveNet, a deep neural network for generating raw audio waveforms. Inspired by recent breakthroughs in neural auto regressive generative models that describe complex distributions such as images and text, this work investigates raw audio generating strategies. State-of-the-art generation is achieved by modelling joint probabilities over pictures or words using neural architectures as products of conditional distributions. Surprisingly, these structures are capable of modelling hundreds of random variables. WaveNet, an audio generating model based on the Pixel CNN architecture, is introduced in this paper. The following are the primary contributions of this work: WaveNets may generate raw voice signals with a level of subjective naturalness never seen before in the field of Text-To-Speech (TTS), as determined by human raters.

2.2. SIGNAL PROCESSING FOR MUSIC ANALYSIS

The system proposed by Meinard Muller et al.,[3] concerned the application of signal processing techniques to music signals, in particular to the problems of analyzing an existing music signal (such as piece in a collection) to extract a wide variety of information and descriptions that may be important for different kinds of applications. The distinctive properties of music audio - such as the predominance of distinct fundamental periodicities (pitches), the preponderance of overlapping sound sources in musical ensembles (polyphony), the variety of source characteristics (timbres) and the regular hierarchy of temporal structures - have shaped a distinct body of techniques and representations (beats).Tempo, beat, and rhythm are all musical elements that are crucial to understanding and

interacting with music. The beat is the continuous pulse that propels music forward and establishes a piece's temporal framework. The beat can be thought of as a series of perceived pulses that are spaced out in time and correlate to the pulse a human taps along when listening to music. The rate of the pulse is then referred to as tempo. Note onsets or percussive events are usually accompanied by musical pulses. Detecting such moments within a signal is a key task known as onset detection.

Gunawan et al., [4] numerous studies of music signal processing have naturally concentrated on monophonic signals. While monophonic signals do produce superior results, the quest for broader applicability has led to a shift in recent years toward the more difficult and realistic situation of polyphonic music. There are two major approaches to dealing with polyphony: the signal can be treated globally, immediately extracting information from the polyphonic signal, or the signal can be divided up into discrete components (or sources) that can then be processed individually as monophonic signals. However, the source separation phase of this latter technique is not always apparent, and it can simply offer a mid-level representation that aids further processing stages. In the parts that follow, we'll go over some fundamentals of source separation before demonstrating several solutions on a variety of music signal processing jobs. The tasks of multi-pitch estimation and musical voice extraction, including melody, bass, and drum separation, are addressed in particular.

Music analysis signal processing is a vibrant and quickly expanding field of research that can enrich the larger signal processing community with new applications and issues. Music is arguably the most complex and meticulously constructed sound signal, and extracting information relevant to listeners necessitates the kinds of specialized methods we've presented, capable of accounting for music-specific characteristics such as pitches, harmony, rhythms, and instrumentation.

2.3. A REVIEW OF RAGA BASED MUSIC CLASSIFICATION AND MUSIC INFORMATION RETRIEVAL (MIR)

This paper is proposed by Kirthika et al., [5] in which the Raga Based Music Classification is described as a component of Music Information Retrieval (MIR) systems for obtaining musical data. Music has been more than just a source of entertainment for us. Musicology researchers, musicians, music learners, music therapists, and other MIR users demand not only the sound but also the actual strength of the music that regulates our bodies, inspires us, treats mental sickness, and so on. Raga is the foundation of all forms of music. A raga is a specific set of notes that follow a set of rules that, when followed attentively, retain and safeguard its purity while creating the classification of ragas in Carnatic, Hindustani, and Western musical traditions is discussed in this study. It also explains how to represent a raga using various attributes gathered from audio data. Finally, presenting a system architecture for music classification based on ragas musical miracles. Audio mining is a method of automatically analyzing and searching the content of an audio source. The various ways of machine learning applications, voice processing, and language processing algorithms are all part of the audio mining study. Even today, looking for desired clips in audio databases is more difficult than searching for wanted words using search engines like Google. To get over this limitation, a lot of effort has gone into developing ways for automatically analyzing and summarizing multimedia content for search and retrieval. Music information retrieval is aided by music processing techniques. Keywords, metadata, handwritten audio tags, and isolated tracks are used to categorize the audio data. Chord estimation melody extraction, frequency estimation and tracking, and structure segmentation are some of the highly trained audio retrieval systems. However, all music lovers and researchers who do not view music solely as enjoyment require a melody or raga-based music/audio retrieval system.

Sturm et al., [9] Data sampling, structural segmentation, onset and offset detection, feature extraction, raga classification, and clustering are among the suggested system's modules. An algorithmic technique for assigning a particular piece of input data to one of a number of clusters or categories is known as classification. To combine song or audio data with comparable ragas, the output of the feature extraction module can be input into any well-known classifier, such as nearest neighbor classifier, SVM classifier, Bayes Classifier, and so on. These clusters can also be utilized to retrieve music information. Briot et al., [1] introduces a relatively new audio mining technique for music information retrieval that emphasizes raga mining. Finally, a computational model for raga-based classification is described, which serves as a starting point for developing an effective MIR system. Mukherjee et al., [2] system's next development will focus on technologies and apps that summarize audio datasets based on ragas and improve current audio search algorithms.

3. SYSTEM DESIGN

Initially a piano song is given as an input and in the analyzation technique, using time domain characteristics, the frequency is detected. Fluctuations are removed in Averaging process. After averaging, detecting all the tall peaks while still preserving the small but significant notes. The most significant frequency component is identified using DFT. Finally, mapping the detected frequencies to its corresponding notes from the notes-frequency table and the output notes as follows.

3.1. SYSTEM ARCHITECTURE

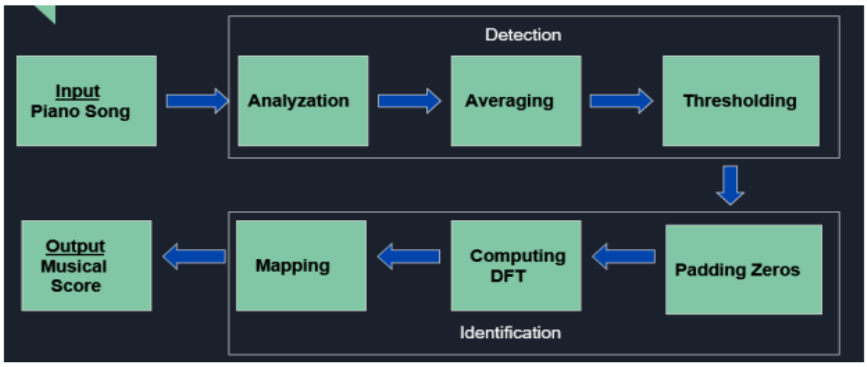

Figure 3.1: Overall system architecture

3.1.1. ANALYZATION

Human can hear signal frequency ranging from 20-20 kHz. From this wide range some part is associated with piano. Different pianos are having different ranges. Each tone of piano is having one particular fundamental frequency and represented by a note like C, D ...etc. The moment we press one note to the immediate other note, the amplitude is initially high enough and decreases with time. If we are able to detect the duration of each note from the time domain characteristics, we can detect and identify the frequency.

3.1.2. AVERAGING

As there is large number of sample for a song and many fluctuations are also present. First step is averaging, where for every 20 samples average is found and the value is assigned to first sample, again for next 20 sample the average value is assigned to 2nd sample. This will not only reduce the number of samples but also remove the fluctuations present.

3.1.3. THRESHOLDING

After averaging we need to detect the peaks from the averaged signal. There are the possibilities when we take some constant threshold value, for some notes it may be higher than the maximum value of the note and for some notes it is low enough such that two peaks of the note get merged and only one peak (one note) could be detected. To overcome this problem the concept of adaptive thresholding came into picture. Using adaptive thresholding, we can detect tall peaks (high frequency notes) while still preserving some of the small but significant notes.

3.1.4. PADDING WITH ZERO

Zero padding is a technique typically employed to make the size of the input sequence equal to a power of two. In zero padding, you add zeros to the end of the input sequence so that the total number of samples is equal to the next higher power of two.

3.1.5. COMPUTING DFT

A Fourier transform provides the means to break up a complicated signal, like a musical tone, into its constituent sinusoids. This method involves many integrals and a continuous signal. We want to perform a Fourier transform on a sampled (rather than continuous) signal, so we have to use the Discrete Fourier Transform instead. After padding zeros, the DFT of the resultant signal is found. Then the corresponding frequency of a particular note is found using equation

$$F = i/T * F_s$$

Where i = index at which maximum amplitude exists, T = Total samples in the fft at a time.

3.1.6. MAPPING

After finding the frequency using DFT, each frequency is mapped with its corresponding note's.

4. IMPLEMENTATION AND RESULT

This method to get the input audio file and plotting the graph with time in X axis and amplitude in Y axis. Here, Fur Elise song in piano version is given as input and the sampling frequency (F_s) has been speeded up by 4 times.

Figure.4.1 shows the plot for analyzation. The X axis represents the time range and the Y axis represent the amplitude of the song. Each piano tone has a single fundamental frequency, which is represented by a note such as C, D, etc. The amplitude is initially high enough when we push one note to the next note and gradually lowers over time.

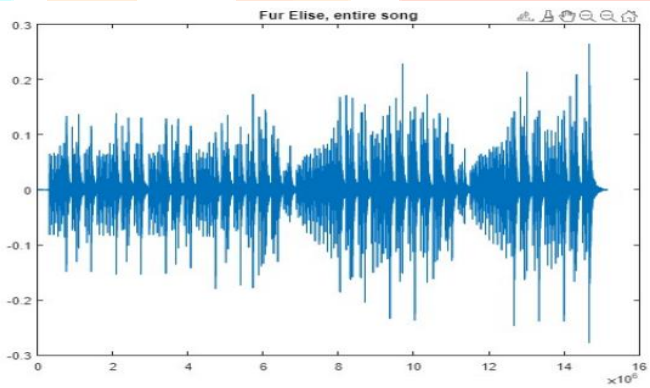

Figure.4.1 Piano note time domain analysis

4.1. ANALYSING THE WINDOW OF A SONG

The window of size ranging from t1 and t2 is gone through the entire song during analyzation. Figure.4.2 shows the plot for analysing the window of the song with the t1 and t2 parameters. The X axis represents the time range and the Y axis represent the amplitude of the song. Windowing is the process of taking a small subset of a larger data, for processing and analysis. The rectangular window, involves simply truncating the audio file before and after the window, while not modifying the contents of the window.

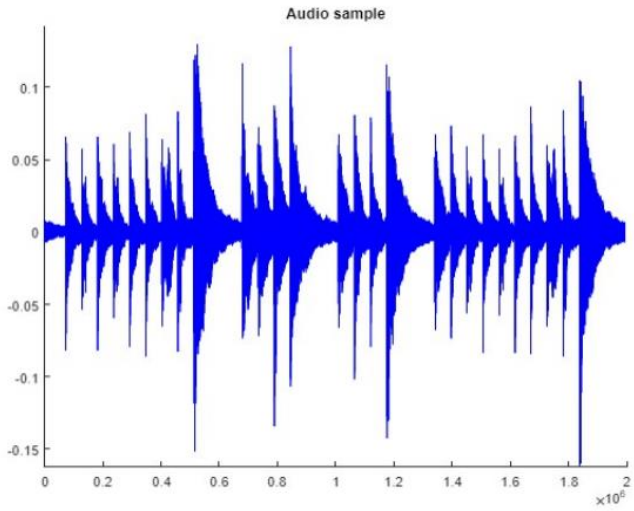

Figure 4.2: Window of the song

4.2. AVERAGING

This method to down sample 20 times using average filter to find out the high frequency content and reduce the fluctuations to speed up the calculations. Figure.4.3 shows the plot for averaging. The X axis represent the samples ranging to 20 and the Y axis represent the amplitude of the song. When the decay in the signal is slow, the averaged signal is more densed. But in case of fast decaying the averaged signal represents the envelope of the original signal.

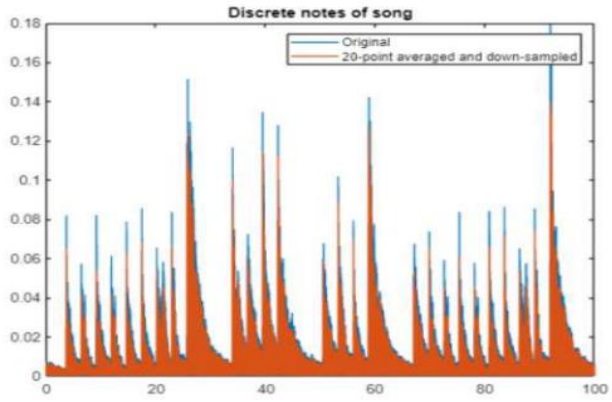

Figure 4.3: Effect of Averaging

4.3. THRESHOLDING

This method to threshold the notes using adaptive thresholding. This way the tall peak notes and small but significant notes are identified. Figure4.4 shows the plot for thresholding to detect the peak notes. The X axis represents the time range and Y axis represents the amplitude of the notes. In constant thresholding one optimum value is decided for which we are able to get maximum number of peaks. When we choose a constant threshold value, it is possible that for some notes it will be larger than the maximum value of the note, while for others it will be low enough that two peaks of the note will be merged and only one peak (one note) will be identified. The notion of adaptive thresholding was introduced to solve this problem.

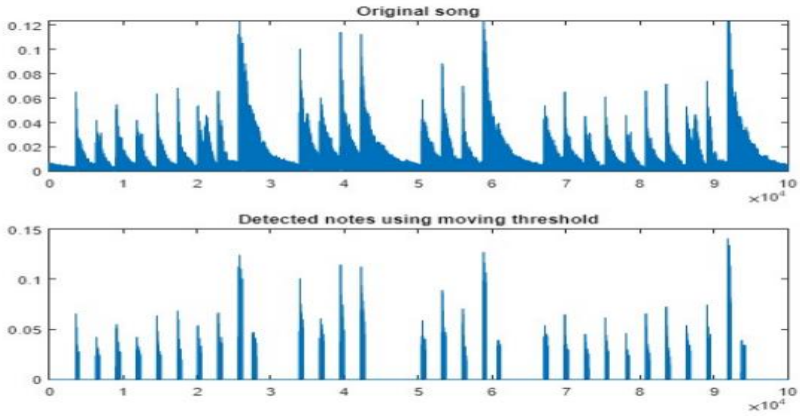

Figure 4.4: Signal after thresholding

4.4. MAPPING

This method for mapping the detected frequencies to its notes. The 108 key notes spanning from C0 to B8 with the frequency range of 16.35 and 7902.13 HZ. Figure.4.5. shows the plot for the fundamental frequency determined using FFT after padding with zeros. The DFT of the resultant signal is found to determine the actual note (Eb4). The fundamental frequency is found to be 313.8035 Hz.

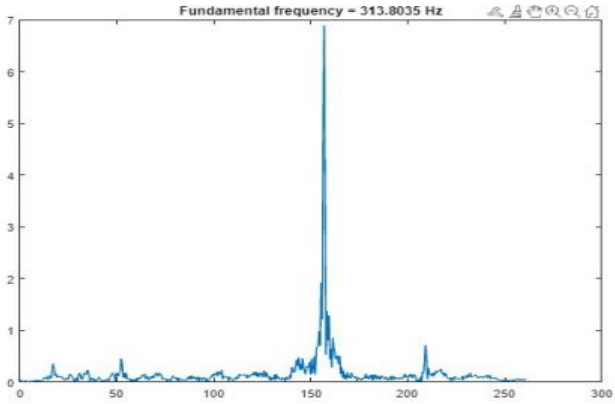

Figure 4.5: Frequency peak for the note “Eb4”

CONCLUSION

The frequencies of a piano tune are recognized in this project, and the matching notes are identified with duration. The note identification approach utilized here is more optimized than earlier methods. We can acquire the desired outcomes with the time duration of each note by altering factors such as threshold values and width. As a result, a project can be used as a learning aid.

This work includes implementation of the virtual piano. The Virtual piano to be used as the evaluation metric for the output of the current system. Also the sheet music could be included. Sheet music is a printed form of musical notation that uses musical symbols to indicate the pitches, rhythms or chords of a song or instrumental musical piece.

REFERENCES

[1] Briot, J. P., & Pachet, F. (2020). Deep learning for music generation: challenges and directions. *Neural Computing and Applications*, *32*(4), 981-993.

[2] Gunawan, A. A. S., Iman, A. P., & Suhartono, D. (2020). Automatic music generator using recurrent neural network. *International Journal of Computational Intelligence Systems*, *13*(1), 645-654.

[3] Kirthika, P., & Chattamvelli, R. (2012, July). A review of raga based music classification and music information retrieval (MIR). In *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)* (pp. 1-5). IEEE.

[4] Kang, S., Ok, S. Y., & Kang, Y. M. (2012). Automatic music generation and machine learning based evaluation. In *Multimedia and Signal Processing: Second International Conference, CMSP 2012, Shanghai, China, December 7-9, 2012. Proceedings* (pp. 436-443). Springer Berlin Heidelberg.

[5] Mukherjee, Himadri, Sk Md Obaidullah, Santanu Phadikar, and Kaushik Roy. "MISNA-A musical instrument segregation system from noisy audio with LPCC-S features and extreme learning." *Multimedia Tools and Applications* 77 (2018): 27997-28022.

[6] Muller, M., Ellis, D. P., Klapuri, A., & Richard, G. (2011). Signal processing for music analysis. *IEEE Journal of selected topics in signal processing*, *5*(6), 1088-1110.

[7] Romagnoli, M., Fontana, F., & Sarkar, R. (2011). Vibrotactile recognition by western and Indian population groups of traditional musical scales played with the harmonium. In Haptic and Audio Interaction Design: 6th International Workshop, HAID 2011, Kusatsu, Japan, August 25-26, 2011. Proceedings 6 (pp. 91-100). Springer Berlin Heidelberg.

[8] Skúli, S. (2017). How to generate music using a lstm neural network in keras. *Towards Data Science*, *7*.

[9] Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., ... & Pachet, F. (2019). Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, *48*(1), 36-55.

[10] Verma, P., & Chafe, C. (2021). A generative model for raw audio using transformer architectures. In *2021 24th International Conference on Digital Audio Effects (DAFx)* (pp. 230-237). IEEE.

[11] Wibowo, H. A. (2019). Generate piano instrumental music by using deep learning. *Режим доступа: https://towardsdatascience.com/generate-pianoinstrumental-music-by-using-deep-learning-80ac35cdbd2e*.

[12] Yamshchikov, I. P., & Tikhonov, A. (2020). Music generation with variational recurrent autoencoder supported by history. *SN Applied Sciences*, *2*(12), 1937.