IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 August 2023, accepted 23 August 2023, date of publication 29 August 2023, date of current version 5 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309639

RESEARCH ARTICLE

Generating Polyphonic Symbolic Emotional Music in the Style of Bach Using Convolutional Conditional Variational Autoencoder

JACEK GREKOW
Faculty of Computer Science, Bialystok University of Technology, 15-351 Bialystok, Poland
e-mail: j.grekow@pb.edu.pl

This work was supported in part by the Bialystok University of Technology under Grant WZ/WI-IIT/3/2023, and in part by the Ministry of Science and Higher Education.

ABSTRACT In times of increasing human-machine interaction, the implementation of emotional intelligence in machines should not only recognize and track emotions during human interaction, but also respond with appropriate emotional content. Machines should be able to react and respond to human emotions. Music generation with a specific emotion is part of this task. This article presents the process of building a system generating polyphonic music content of a specified emotion using a conditional variational autoencoder and convolutional layers. The process of preparing a database of training examples with compositions by Johann Sebastian Bach, selecting and conducting transformations of musical examples was described. Annotation with emotion labels was done by music experts with a university music education. The four emotion labels - happy, angry, sad, relaxed - corresponding to the four quadrants of Russell's model were used. The process of coding symbolic music examples into a time-pitch matrix representation, but also the structure of the built variational autoencoder, was described. Experiments on the implementation of different convolutional layers intended for visual analysis of the representation of music examples were presented. The generated emotional music files were evaluated using metrics and expert opinions.

INDEX TERMS Music emotion, polyphonic music generation, symbolic music, variational autoencoder.

I. INTRODUCTION

Machines have been accompanying us in everyday life for a long time. Sometimes we were aware of it, more often not. The human-machine interaction intensifies more and more every year, raising the demands placed on machines. From smart bands to smart homes, man expects a more and more appropriate response, despite the fact that ''it's just a machine.'' Even though a simple reaction is formally easily achievable with clear orders or queries by the circling artificial intelligence, man still has the right to long for what in the human world we call emotional intelligence. In this, it is not enough to simply recognize the current human emotion, even if a multimodal approach is used. For proper, comprehensive communication, an empathetic response is also necessary [1]. Such a non-verbal response can be, depending on the mood or situation, generated music.

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Turchet.

Generating music using machine learning models is a new phenomenon entering the realm of creative human creativity. New systems have been created that imitate human creativity and learn from musical examples created by the greatest composers in the history of music [2], [3]. Music is one of the most abstract arts; it expresses human ideas in the form of sounds organized in time. When analyzing the pitches of a polyphonic piece of music, they can be visualized in the form of two-dimensional graphic images [4], where the horizontal axis is time and the vertical axis is the pitch of the played sounds at a given time. Due to the similarity of images and music represented by two-dimensional graphics, the issues of image generation and music generation may have similar technological solutions based on convolutional layers.

One of the main reasons why we listen to music is the perception of emotions [5]. Depending on the melody, dynamics, and harmony changing over time, we can perceive different emotions. Adding the element of emotion to music-generating systems gives us additional control over

the created content. Similar systems can be used in machine-human communication, where not only emotion recognition is expected but also response generation with the appropriate emotional tone.

The aim of this paper was to present the process of building a model that generates polyphonic music sequences with a specified emotion using a variational autoencoder (VAE). The designed model should learn the musical elements from the training set that affect emotion and apply them when generating new examples. The main contributions of our study are as follows:

- a MIDI music dataset with emotion labels annotated by music experts with a university music education was created;
- a model generating polyphonic music sequences with emotion using variational autoencoder and convolutional layers was proposed, which - to our knowledge - has not been done in other works for generating emotional music;
- a special construction of the convolutional layers for learning the visual representation of symbolic music was proposed.

The rest of this paper is organized as follows. In Section II, we discuss the existing work on music generation, music emotion recognition, the generation of emotional music, and using a variational autoencoder as a generative model. Section III describes the music dataset and the process of transformation and annotation with emotion of the MIDI files. Section IV presents the piano-roll representation of symbolic music and Section V shows the idea of the variational autoencoder and its implementations. Section VI presents the evaluation using metrics and expert opinions. Finally, Section VII summarizes the main findings.

II. RELATED WORK

A survey of various tasks related to symbolic music generation using deep learning was presented by Ji et al. in [6]. The work also presents the music representations used, evaluation methods, popular datasets as well as highlights current challenges. The authors noticed that music generation with a specific emotion is one of the future directions of research development. A functional taxonomy for the key concepts that form the functional goals of music generation systems was presented in the work by Herremans et al. [7]. Zhao et al. in [8] conducted a comprehensive overview and analysis of recent intelligent music generation techniques. Issues raised in the paper concerned music encoding, datasets, comparing generation algorithms, and the existing methods for evaluation.

Most neural network models for music generation use recurrent neural networks, but there are exceptions. Yang et al. [9] used convolutional neural networks and generative adversarial network (GAN) for generating a symbolic melody. They proposed a conditional mechanism to exploit the available prior knowledge so that the model can generate melodies from scratch by following a chord sequence, or by conditioning on previous bars.

Huang et al. [10] used CocoNet, a deep convolutional model with blocked-Gibbs sampling algorithm, for completing partial scores in corrupted symbolic Bach chorale. In the conducted experiments, a random subset of notes were removed from Bach chorale, and the model was asked to infer their values. New note values were sampled from the probability distribution put out by the model. Agostinelli et al. [11] presented MusicLM, a model for generating high-quality music at 24 kHz from text descriptions. The proposed system can be conditioned on a text and a melody, and it can transform input humming or whistling melodies according to the style given in the text. The system used three models (SoundStream, w2v-BERT, MuLan) for extracting audio representations that will serve for conditional music generation. Also a hierarchical sequence-to-sequence modeling task, where each stage is modeled autoregressively by a separate decoder-only Transformer, was proposed.

A. MUSIC EMOTION RECOGNITION

This paper is devoted to the generation of emotion-controlled music, but the opposite task to ours could be the recognition of emotions in music, which is part of the research field in music information retrieval. Both tasks have common areas, such as the emotion model, audio features connected with music emotions, as well as problems with different perceptions of emotions.

In papers devoted to music emotion recognition division into categorical and dimensional with regard to the emotion model approach can also be found [12]. In the categorical approach, a number of emotional adjectives are used for labeling music excerpts [13], [14], [15]. In the dimensional approach, emotion is described using dimensional space, like the 2D model proposed by Russell [16], where the dimensions are represented by arousal and valence [17], [18], [19].

In [20] content-based music emotion recognition was presented as a classification and regression problem, which was closely connected with the selected emotion model - categorical and dimensional, respectively. The author focused on examining audio as well as MIDI files and for each presented the relevant feature sets that describe them. Due to the fact that emotion in music can change over time, and can be constant only in short excerpts, emotion maps created from music that visualize emotion distribution over time were proposed. Panda et al. in [21] presented relations between eight musical dimensions (melody, harmony, rhythm, dynamics, tone color, expressivity, texture, and form) and specific emotions. Authors also reviewed the emotionally-relevant computational audio features from four common audio frameworks (Marsyas, MIR Toolbox, PsySound, and Essentia) used in music emotion recognition. In [22] the authors investigated to which extent state of the art machine learning methods are effective in classifying emotions in the context of individual musical instruments, and how their performances

compare with musically trained and untrained listeners. In the experiments, four emotions (aggressiveness, relaxation, happiness, and sadness) with three emotion intensity levels (low, medium, high) were used and the dataset contained classical and acoustic guitar excerpts. The results showed that emotions were better recognized by musicians rather than listeners with no musical background with respect to the original intention of the composer. By classifying emotions, the machine perception of emotions matched or exceeded human performance for three out of four emotions, except for the emotion relaxation.

B. EMOTION-BASED MUSIC GENERATION

The generation of emotion-controlled music is still in its early stages of development and the collection of works devoted to this topic is not yet very rich [23]. Papers concerning emotion-based music generation present the use of different deep learning models, different training data, as well as different emotion models.

The first overview of systems for algorithmic composition with the intention of targeting specific emotional responses was created by Williams et al. [24]. The idea of polyphonic music generation with a specified positive or negative emotion was presented by Ferreira and Whitehead [25]. They used a single-layer multiplicative long short-term memory (mLSTM) network, which developed the method used for generating textual product reviews with a sentiment. The model was controlled by optimizing the weights of the found neurons responsible for the sentiment signal. The training dataset was collected from video game soundtracks in MIDI format. Madhok et al. [26] presented a framework that generates relevant music based on the emotion detected from a person's facial expressions. The music generation model was constructed with LSTM architecture and the emotion model used seven categories - angry, disgust, fear, happy, sad, surprised, and neutral. Zhao et al. in [27] used Bi-axial LSTM networks to generate polyphonic music. The proposed solution generated polyphonic examples with one of the four emotions - happy, tensional, sad, peaceful. The authors trained the model with a global condition of emotional vectors and design tunable parameters for generating music of a corresponding emotion. Hung et al. in [28] presented the Transformer and LSTM models for emotion conditioned symbolic music generation using a multi-modal (audio and MIDI) database, which consisted of pop piano music labeled with four classes of perceived emotions. The emotion labels correspond to four quadrants of Russell's model. An approach for the generation of multi-instrument symbolic music driven by musical emotion was presented by Sulun et al. in [29]. The solution used different conditioning from the Transformer and used a symbolic music dataset annotated by continuous arousal and valence values. The use of a Transformer to generate music with a controlled emotion was proposed by Pangestu et al. in [30]. Emotions were described using three categories—negative, neutral, and positive. Neves et al. in [31] proposed a generative model of symbolic music conditioned by emotion. The trained model consisted of Transformer-GAN and emotion labels were in the form of continuous values of valence and arousal.

C. VAE AND MUSIC GENERATION

Use of a generative model with the architecture of a variational autoencoder (VAE) has advantages in music generation because of the ability to control the generation process with a latent variable. In the survey [32], Zhang analyzed representation learning methods for controlled music generation. He explained how a musical fragment can be abstracted and reduced to one or several representations, such as rhythm, chords, or emotion. By controlling the representation of music, humans can control the process of generating music. Several models based on VAE and on disentanglement learning and hierarchical structure learning were presented.

Roberts et al. in [33] used a recurrent VAE that utilizes a hierarchical decoder for improved modeling of sequences with a long-term structure. The constructed model was tested on symbolic MIDI data in the form of monophonic melodies, drum patterns, and trio sequences consisting of separate streams of a melodic line, a bass line, and a drum pattern. Wang et al. in [34] proposed a VAE framework, with latent vectors representing chords and style of polyphonic symbolic music. The trained network was used in tasks such as compositional style transfer, style variation, and accompaniment arrangement. In [35], Guo et al. used a generative VAE model to control tonal tension in the generated music. For identifying latent tension variables, the labeled musical fragment positions in the latent space were calculated. The generated music is similar to the original music by keeping the rhythm and manipulating the pitches to match the tonal tension.

What distinguishes this work from others is that it uses a conditional VAE with the emotion parameter influencing the generated polyphonic music examples. We investigated the structure for the convolutional layers in VAE encoder and decoder for encoding and decoding visual representations of music examples.

III. TRAINING DATA

A. SYMBOLIC MUSIC DATASET

In this work, the music21 library [36] containing compositions by Johann Sebastian Bach was used. In this library, the content of the compositions is saved in a symbolic form, which means that we have access to sound parameters such as pitch, length, volume, etc., and we do not have to decode them from the audio files. The collection includes mostly chorales (382) as well as other compositions, for a total of 410 pieces. The list of all compositions is available in [37] in MusicXML format.

To use music21's collection of compositions to train the model for generating polyphonic sequences, several transformations were undertaken (Fig. 1). The first transformation was to equalize the note duration. The note duration in a

composition is affected by beats per minute (BPM), tempo, and note type. Due to the fact that the compositions in the database were saved at different tempos, the tempos of all songs were standardized to 120 BPM, or 120 quarter notes per minute. The note values of songs with a tempo other than 120 BPM were corrected. Thus, a dataset was obtained in which only the note types - sixteenth note, eighth note, quarter note, half note, whole note - affected the length of the notes.

The second dataset transformation consisted of limiting the length of the musical example to four bars and selecting only compositions with the time signature 4/4, which are the majority in music21. This resulted in a slight reduction in the number of examples in the database. This way, the rhythmic structure of the examples was unified and eventually contained four bars with a 4/4 time signature. As a result, eight seconds of music examples were obtained, each with a tempo of 120 BPM.

The third transformation concerned the compositions' keys, which are different in the examples from the music21 database. The same melody in different keys sounds similar, and a training set in different keys would make the task of training even more difficult. To facilitate the training, all the compositions were transposed to C major or C minor. Thus, we assumed that our model would generate polyphonic sequences in C major and C minor scales. After the performed transformations, a unified dataset was obtained, with 338 polyphonic sequences of uniform 8 s. length. All the transformed examples were stored in MIDI format.

The preprocessing methods used in this work were used in other papers on symbolic music generation. Discretization of music data with sixteenth notes simplifies note value coding and were used in [10], [27], and [38]. Also limiting the length of the musical examples in the training dataset was used in [39]. A data usage limitation that only uses music with a 4/4 time signature was used in [27], [30], and [39]. The transposition of all examples into a single common key as preprocessing is not used often. This solution was used in a polyphonic music generation system in [40]. Due to the model of emotions with negative and positive emotions, our solution proposes a transposition into two keys - major and minor - which was used intentionally to simplify the training of the model.

B. EXAMPLE ANNOTATION WITH EMOTION LABELS

To train the machine learning model to generate musical sequences with a specific emotion, it was necessary to label the music examples with emotion labels. During the annotation, four emotion labels were used - happy, angry, sad, relaxed - corresponding to the four quadrants of Russell's model [16] Q1-Q4. In Russell's model (Fig. 2), emotions are distributed on a plane divided by two parallel axes - arousal and valence. Arousal can be high or low and valence positive or negative. The labels used indicate a group of emotions in a given quadrant, e.g. the label happiness refers to a group of different emotions in quadrant Q1, where arousal is high and valence is positive. A similar division of emotions into four

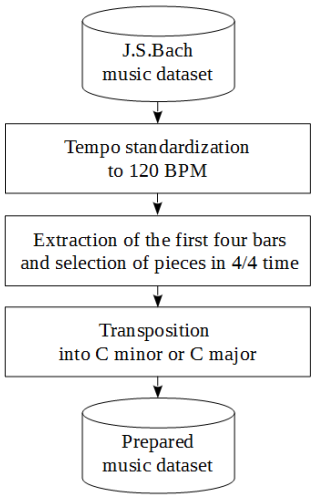

FIGURE 1. Transformations of music dataset.

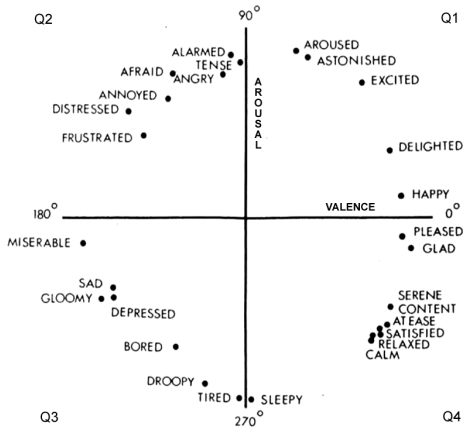

FIGURE 2. Russell's circumplex model [16].

basic categories was used, among others, in [27], [28], and [41].

The music files were played at one timbre (MIDI instrument: Grand Piano) and one volume, so the timbre and volume did not affect the emotions. What did affect the emotions in the musical fragment was the pitch, length, rhythmic arrangement of sounds, the harmonic relationships between them, and the major/minor scale.

When annotating the music examples, one can refer to the felt or perceived emotions [42]. The felt emotion is the one that the listener feels at a given moment - e.g. if he listens to something very sad, he communicates that the emotion is sad and at the same time, for example, he wants to cry. The perceived emotion is the one that the listener notices in the song but does not physically succumb to it - e.g. he listens to something sad, communicates that the emotion is sad, but he does not want to cry. In our experiment, the music experts were tasked with labeling MIDI songs with the perceived emotions.

TABLE 1. MIDI files annotated with four emotions.

Emotion	Abbr.	Quarter in Russell's model / Arousal-Valence	Files in major/minor scale	Amount of files
happy	e1	Q1 / high-high	89/1	90
angry	e2	Q2 / high-low	2/87	89
sad	e3	Q3 / low-low	6/71	77
relaxed	e4	Q4 / low-high	81/1	82

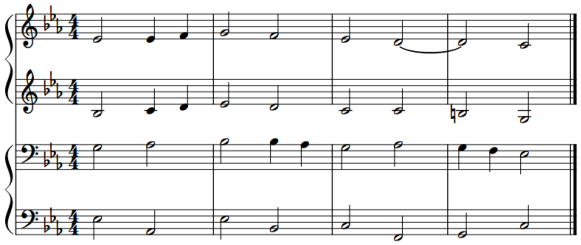

FIGURE 3. Fragment of sheet music of Chorale BWV 96 by J.S.Bach.

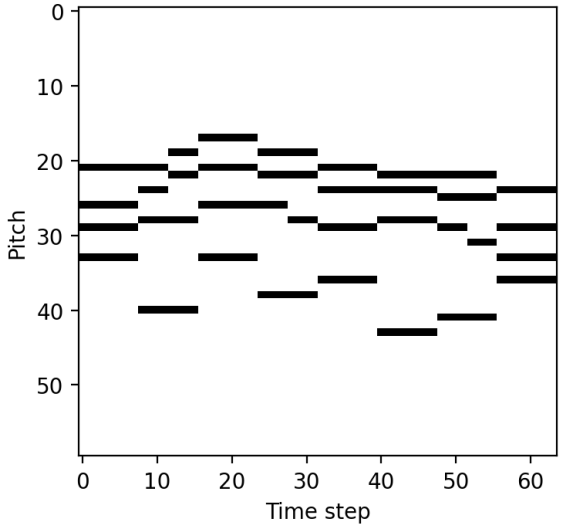

FIGURE 4. Piano-roll representation of a sheet music fragment of Chorale BWV 96 by J.S. Bach.

The labeling was done by five experts with a university music education. The opinions of music experts - people who play in bands, compose and interpret music on a daily basis - are more reliable than people who only work with music occasionally. Each of the music experts labeled all 338 MIDI files, which gave them an overview of the entire music collection and the shades of emotions within, which is not always maintained when labeling music databases. Labeling the entire set by each expert has a positive effect on the quality of the annotations, which was emphasized in [43].

The data collected from the five experts were averaged. Investigating the internal consistency of the collected data, Cronbachs α obtained a value of 0.88, which confirmed good annotation consistency. The number of files labeled with the four emotions is shown in Table 1. We can see that almost all examples with positive emotions - e1 and e4 (high valence) - are in the major scale and examples with negative emotions - e2 and e3 (low valence) - in the minor scale. The entire set of MIDI files labeled with the emotions along with the proposed system code and generated music examples can be found at the following link.[1]

IV. CODING MUSIC EXAMPLES

Due to the fact that the music generating system would learn from polyphonic pieces, it was decided to encode all MIDI files from the database using piano-roll representation. In piano-roll representation, the horizontal axis describes the time steps of the duration of the music example, and the vertical axis indicates the pitches of the notes that are switched on and off. Fig. 3 presents the notation of a fragment of Chorale BWV 96 by J.S. Bach from our database of examples and the corresponding piano-roll representation is shown in Fig. 4. The MusPy Toolkit [44] was used to read the MIDI files and convert them to piano-roll representation.

[1]https://github.com/grekowj/musgenvaecnn_4v

Piano-roll representation encodes music in a time-pitch matrix, where the columns are time steps and the rows are pitches. The values in the matrix indicate the presence of sounds at different time steps. The shape of the standard matrix is $T \times 128$, where T is the time step number. In the MIDI format, the possible pitches are 0-127, hence the number of rows in the matrix is 128.

The length of each example in the database corresponds to four bars in a 4/4 time signature, which is equal to four quarter notes per bar, for a total of 16 quarter notes. The shortest note in the database is a sixteenth note, and thus the music examples were encoded (discretized) with a time step corresponding to a sixteenth note. There are four sixteenth notes for each quarter note, so dividing the entire music example by the shortest note (the sixteenth note) we get $T = 64$ time steps, 4 (bars) $\times$ 4 (quarter notes) $\times$ 4 (sixteenth notes).

After analyzing all the examples from the MIDI files, it appeared that very high and very low sounds were not used, which allowed to reduce the window of possible pitches to 60 sounds. As a result, we obtained the shape of the output tensor representing the music example 64 $\times$ 60 (time step $\times$ pitch). An example visualization of the time-pitch matrix is presented in Fig. 4. The obtained matrices can be interpreted as visual representations of music examples - i.e. images - and use neural networks to process the images.

V. SYSTEM CONSTRUCTION

A. MODEL

A conditional variational autoencoder (CVAE) was used as a generative model [45]. It encodes the input data (music representation) into latent space with Gaussian distribution and then decodes samples from the latent space into a form similar to the input data (Fig. 5). A property of the trained VAE is that the latent space is continuous and can be navigated by generating new data. In CVAE, on top of the input of the

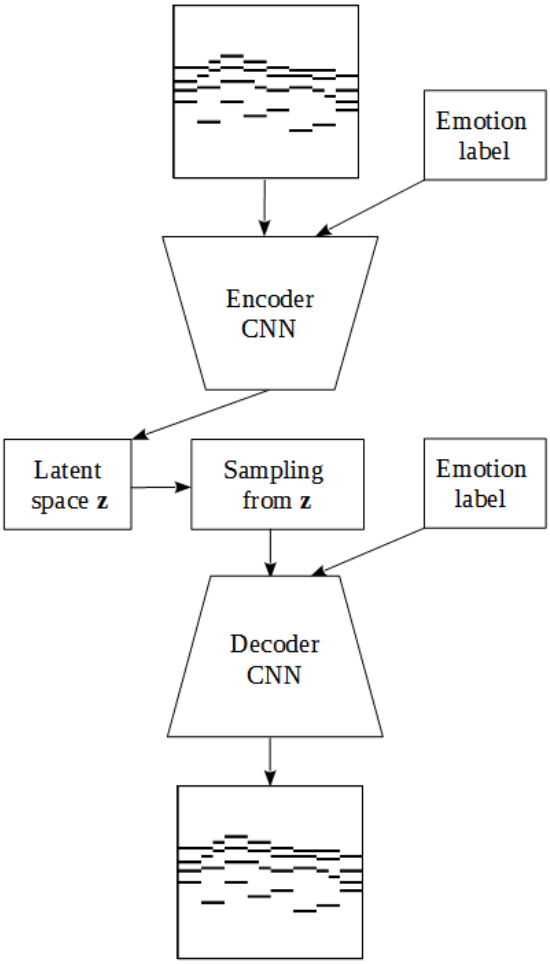

FIGURE 5. Training of the CVAE model.

encoder and decoder, we have an additional condition - the emotion label - which allows to control the emotion of the generated music examples. Only the module of the decoder is used to generate new examples (Fig. 6). The input is given an emotion label and a random sample with the size of the latent space.

The CVAE network consists of the encoder and the decoder joined together. The encoder takes input x, and estimates the mean μ, and the standard deviation σ of the multivariate Gaussian distribution of latent vector z. The decoder takes the samples from latent vector z to reconstruct the input on the output as $\tilde{x}$. The loss function is the sum of both the *Reconstruction loss* ($\mathcal{L}_R$) and *Latent loss* ($\mathcal{L}_L$). *Reconstruction loss* calculates the difference between input x and output $\tilde{x}$ using cross entropy. *Latent loss* is calculated using the Kullback-Leibler divergence, which calculates the distance between the the Gaussian distribution and the actual distribution in latent vector z:

$$\mathcal{L}_L = -\frac{1}{2}\sum_{i=1}^{K}(1 + \log\sigma_i^2 - \sigma_i^2 - \mu_i^2) \qquad (1)$$

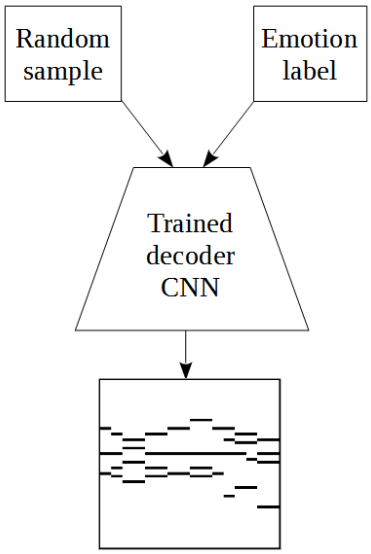

FIGURE 6. Generating new music examples using a trained decoder.

where K is the dimensionality of latent vector z, μ_i and σ_i are mean and standard deviation of i dimension of latent vector z.

B. IMPLEMENTATION

A special construction of the convolutional layers was proposed to analyze the piano-roll representation images. The Keras [46] and Tensorflow[2] deep learning libraries were used to implement the generative models in Python.

Due to the fact that the proposed CVAE is dedicated to the analysis of music representations and consists of two parallel convolutional branches, it was called CVAE-Mus2. The construction of the convolutional parts of the encoder is presented in Fig. 7. It consists of two parallel branches of analysis, which at the end are connected with the Concatenate layer. Each branch contains two sequential convolutional layers (Conv2D) with an increasing number of filters (64, 128) and with special kernel size (1, 12) for analyzing 12 octave semitones, and kernel size (4, 1) for analyzing the next time steps.

Fig. 8 presents CNN layers of the decoder, which, similarly to the encoder, also consists of two parallel branches, sequentially transposed convolution layers (Conv2DTranspose); the number of convolutional filters decreases (64, 128) and the kernel sizes are analogous to those in the encoder (1, 12) and (4, 1). In standard convolutional layers, kernels in the form of a square are most often used, e.g. (3, 3) or (5, 5). Due to the fact that the dimensions of the analyzed images have a certain meaning (pitch, time step), it was decided to modify the kernel size. Kernel (1, 12) analyzed the whole octave (12 semitones) and kernel (4, 1) analyzed four time steps equal to one quarter note. The ReLU function in the decoder and the LeakyReLU function in the encoder were used as the

[2]https://www.tensorflow.org

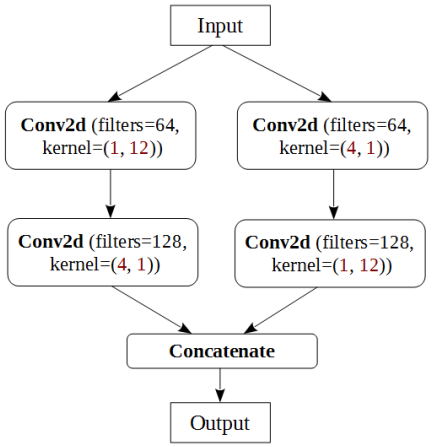

FIGURE 7. Encoder CNN layers of CVAE-Mus2.

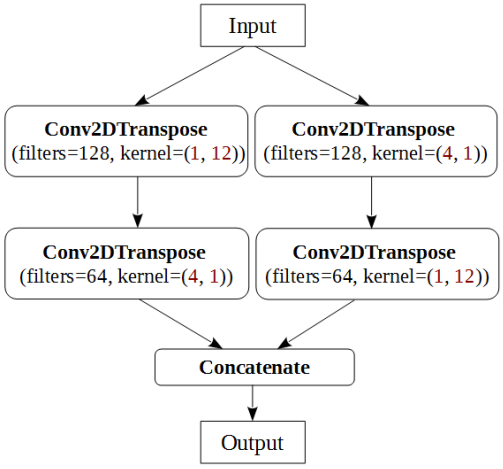

FIGURE 8. Decoder CNN layers of CVAE-Mus2.

activation functions in the layers. The strides of the Conv2D and Conv2DTranspose layers were the same as kernel size.

The detailed construction of the encoder is presented in Fig. 9. We note that the input tensor representing the music example with the shape (None, 64, 60, 1) is concatenated with the condition, which in our case is the emotion label. Next, we see two parallel CNN branches with outputs (None, 16, 5, 128) that are concatenated. Next we have a layer that flattens the tensor to one dimension and two dense layers that reduce dimensionality and generate the mean and log variance. The last output layer of the encoder is a sampling of latent vector z with a shape (None, 32). Similarly, a decoder that takes the samples from latent vector z to reconstruct the piano-roll representation with a shape (None, 64, 60, 1) is created.

During the experiments, the proposed model (CVAE-Mus2) was compared with the baseline model (CVAE-Base), which instead of two parallel CNN branches contained two sequentially connected convolutional layers with a number of filters - 256 and 128, respectively - and kernel size (3, 3) in the encoder (Fig. 10a) and similarly two transposed convolution layers (Conv2DTranspose) in the decoder. Experiments were also conducted on an intermediate model (CVAE-Mus1), which consisted of two sequentially connected convolutional Conv2D layers with 256 and 128 filters, respectively, with modified kernel sizes (1, 12) and (4, 1) (Fig. 10b) and two analogous transposed convolution layers in the decoder. This model differed from the proposed model in that it had only one branch of piano-roll representation analysis.

TABLE 2. Losses and number of trainable params of the tested models.

Model	Loss	Reconstruction loss	Latent loss	Trainable params
CVAE-Base	290.44	230.37	60.06	2.31M
CVAE-Mus1	187.14	128.32	58.82	2.37M
CVAE-Mus2	**183.44**	**126.15**	**57.29**	2.24M

In total, the experiments were done on three models - the base CVAE-Base, the intermediate CVAE-Mus1 and the proposed CVAE-Mus2. The prepared models were trained with the Adam optimizer [47], $1e$–3 learning rate, batch size equal to 4 and 70 epochs. The hidden layer size was 32.

Table 2 presents the final training losses and number of trainable params of the tested models. The number of trainable params of the tested models is at a similar level. We note that the CVAE-Mus2 model performed better compared with the baseline model (CVAE-Base) and the intermediate model (CVAE-Mus1). It had a lower reconstruction loss (126.15), and the hidden layer distribution is more similar to the Gaussian distribution than the other models (less hidden layer loss). We can also see a clear loss reduction for the CVAE-Mus1 and CVAE-Mus2 models compared with the CVAE-Base model.

VI. EVALUATION OF THE GENERATED EXAMPLES

A. EVALUATION USING METRICS

To evaluate the generated files with a specific emotion, metrics [44] that analyze the musical sequence in terms of pitch, their number, use of sounds from a given scale, etc. were used. The following metrics were calculated:

- *pitch range* - defined as the difference between the highest and the lowest pitch;
- *pitch in scale C major rate* - defined as the ratio of the number of notes in the C major scale to the total number of notes;
- *pitch in scale C minor rate* - defined as the ratio of the number of notes in the C minor scale to the total number of notes;
- *polyphony rate* - defined as the ratio of the number of time steps with multiple pitches to the total number of time steps.

To evaluate the generated music examples, training data was selected as a reference point and the generated examples were compared with this training data. Using the decoder from the training model, 20 music examples were generated for each of the four emotions, i.e. 80 examples for each of the

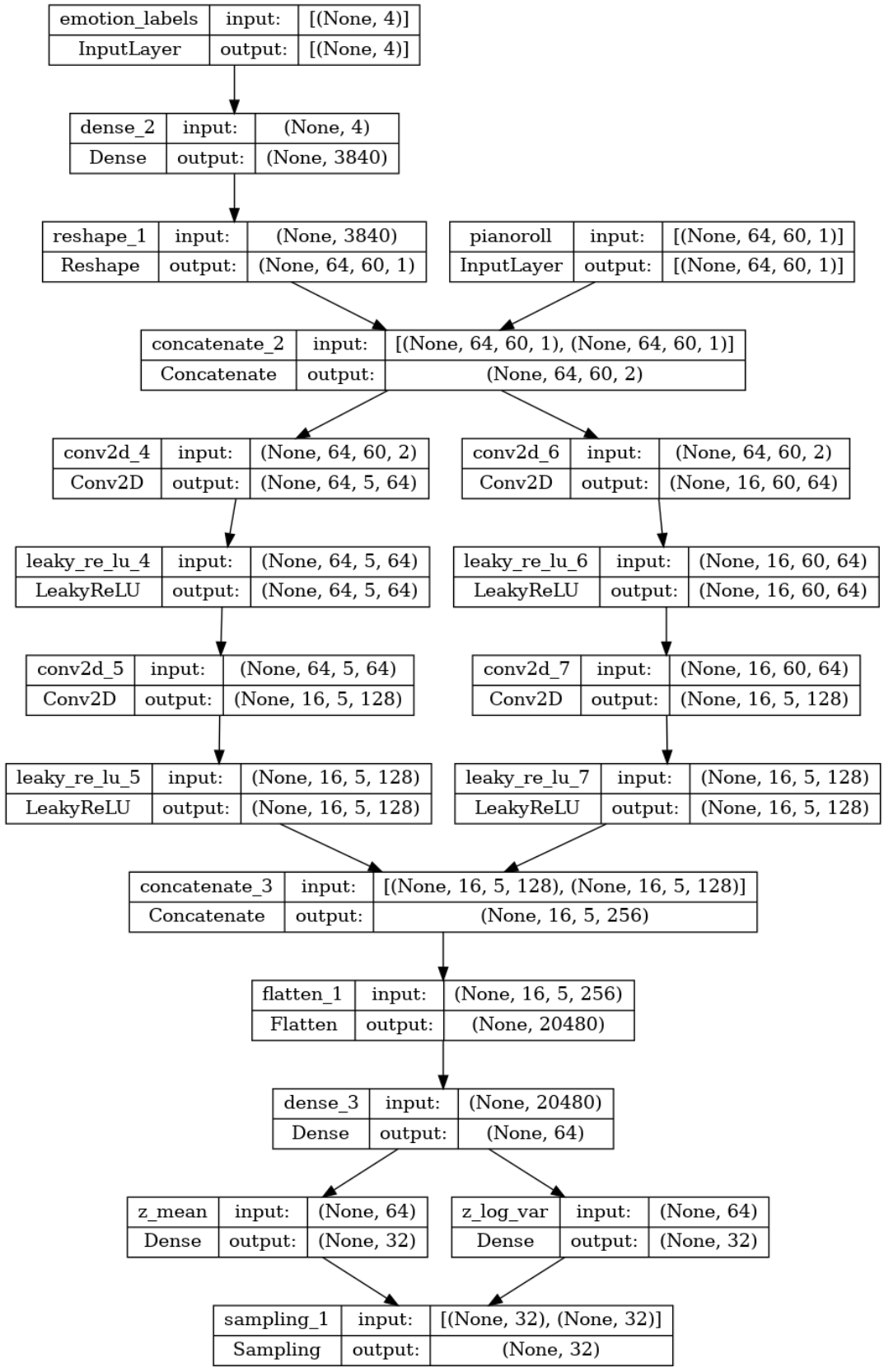

FIGURE 9. Encoder of CVAE-Mus2.

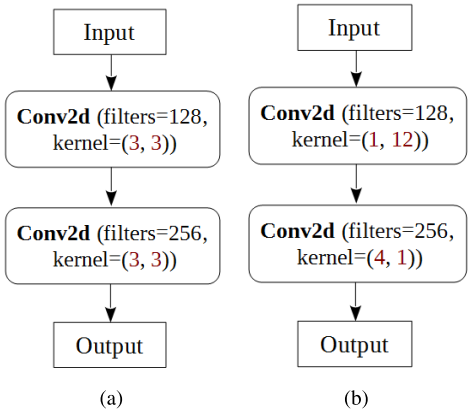

FIGURE 10. Encoder CNN layers of CVAE-Base (a) and CVAE-Mus1 model (b).

tested models. Selected examples of the generated music files can be found and played at the following link.[3]

Four metrics were calculated for each generated polyphonic sequence as well as for each file from the training set. Table 3 presents the calculated means (μ) and standard deviations (σ) of the metrics obtained for the music with four emotions (e1, e2, e3, e4) generated using CVAE-Mus2, CVAE-Mus1 and CVAE-Base models, as well as the music used for the training models. The results from the generated examples that are closer to the results from the training set are marked in bold.

We noted that the mean values (μ) of the metrics obtained for models CVAE-Mus2 and CVAE-Mus1 were closer to the metrics from the training set than the metrics for the baseline model. CVAE-Mus2 won in most cases (11/16) and CVAE-Mus1 was better only in five out of 16 cases. This confirmed the use of a special kernel size for piano-roll representation analysis. Based on the collected statistics, it can be concluded that the proposed model better recognized patterns in files labeled with emotions than the baseline and intermediate models, which generated files with characteristics more similar to the training data.

It appears that the differences between the mean metric values for individual emotions were smaller for the metrics *pitch in C major scale rate* and *pitch in C minor scale rate* than for *pitch range* or *polyphony rate*. It can be concluded that the proposed model better learned to use the sounds of two opposing major and minor scales to apply them when generating music sequences.

Fig. 11 presents the distributions of the *pitch range* in the form of a box plot for the generated and the training sets labeled with emotions e1-e4. We note higher values for the generated sets then for the training sets, which shows that the task of generating files with the correct *pitch range* was difficult. Among the tested models, values for CVAE-Mus2 better reflect the characteristics of the training set - slightly higher median values for emotions with high arousal (e1, e2) and slightly lower median values for emotions with low arousal (e3, e4). It can be seen that the problem of generating files especially for low arousal emotions (e3, e4), where the music is calmer and the *pitch range* values should be smaller. Analyzing the distributions for the CVAE-Base, CVAE-Mus1, and CVAE-Mus2 models, we notice a tendency to approach the training set; the median values of the *pitch range* for all emotions gradually decreased for models CVAE-Mus1 and CVAEMus2, approaching the values of the training set.

The distributions of the *pitch in C major scale rate* are shown in Fig. 12. Generally, the major scale is associated with positive emotions (e1, e4 - positive valence) and the minor scale with negative ones (e2, e3 - negative valence). This can be seen in the higher values of the *pitch in C major scale rate* for emotions e1 and e4 and lower values for e2 and e3. We can see how these metrics for CVAE-Mus2 and CVAE-Mus1 come close to the metrics calculated for the training data. The furthest are the metrics for the generated set with the CVAE-Base model.

Fig. 13 presents the distributions of the *pitch in scale C minor rate*. We see the rule of using the notes of the C minor scale in the generated and the training sets: lower values for emotions e1 and e4 (positive valence) and higher values for e2, e3 (negative valence). Analyzing the distributions of the *pitch in scale C minor rate* and *pitch in scale C major rate* (Fig. 12), it can be stated that all the tested models learned how to generate files with emotions with a similar distribution as the training files. CVAE-Mus2 and CVAE-Mus1 were the better models, while the CVAE-Base model was only slightly worse. The reason that all three models achieved good results is probably that all the polyphonic sequences in the training set were in C major and C minor scales, which are associated with positive and negative valence, respectively. The transformation of all music files into two scales resulted in an apt simplification of the task of generating music with positive and negative emotions.

The distributions of the *polyphony rate* are shown in Fig. 14. Analyzing the distribution for the CVAE-Base, CVAE-Mus1, and CVAE-Mus2 models, we note the tendency to approach the training set; the median values of the *polyphony rate* for all emotions gradually increased towards the values of training sets. We also note lower median values for emotions with high arousal (e1, e2) than for emotions with low arousal (e3, e4). It can be said that for the tested models it is easier to generate files with a greater polyphony for calmer emotions. The characteristics of the files generated by CVAE-Mus2 compared with CAVE-Base and CVAE-Mus1 are the closest to the training set.

B. EVALUATION USING EXPERT OPINIONS

As a second method of evaluating the generated music we asked five music experts with a university music education to annotate the emotions of the generated music files. Assessment of the generated examples pertained three models - the baseline (CVAE-Base), intermediate (CVAE-Mus1),

[3]https://grekowj.github.io/research/musgenvaecnn_4v/

TABLE 3. Metrics obtained from the generated and training sets labeled with four emotions.

Metric	Emotion	Generated set model CVAE-Base μ (σ)	Generated set model CVAE-Mus1 μ (σ)	Generated set model CVAE-Mus2 μ (σ)	Training set μ (σ)
Pitch range	e1	39.45 (8.33)	**32.45** (5.45)	36.55 (8.32)	31.71 (3.65)
	e2	34.50 (6.70)	34.40 (10.50)	**33.60** (7.95)	30.70 (2.57)
	e3	38.25 (5.97)	**32.45** (9.14)	37.20 (11.92)	27.73 (3.25)
	e4	40.35 (7.54)	38.10 (9.94)	**36.20** (11.28)	27.04 (4.96)
Pitch in scale C major rate	e1	0.99 (0.02)	0.99 (0.02)	**0.97** (0.03)	0.96 (0.04)
	e2	0.48 (0.09)	0.69 (0.10)	**0.68** (0.11)	0.68 (0.09)
	e3	0.50 (0.10)	**0.68** (0.13)	0.66 (0.10)	0.70 (0.10)
	e4	0.97 (0.04)	**0.98** (0.03)	0.96 (0.06)	0.98 (0.03)
Pitch in scale C minor rate	e1	0.52 (0.06)	0.66 (0.09)	**0.65** (0.08)	0.61 (0.06)
	e2	0.96 (0.05)	**0.94** (0.05)	0.97 (0.03)	0.89 (0.08)
	e3	0.92 (0.06)	0.95 (0.05)	**0.91** (0.08)	0.88 (0.09)
	e4	0.57 (0.07)	0.69 (0.06)	**0.61** (0.06)	0.63 (0.06)
Polyphony rate	e1	0.35 (0.17)	0.51 (0.18)	**0.59** (0.22)	0.99 (0.05)
	e2	0.46 (0.11)	0.52 (0.20)	**0.58** (0.18)	0.99 (0.03)
	e3	0.56 (0.15)	0.59 (0.20)	**0.65** (0.25)	1.00 (0.01)
	e4	0.61 (0.18)	0.54 (0.20)	**0.77** (0.13)	0.97 (0.11)

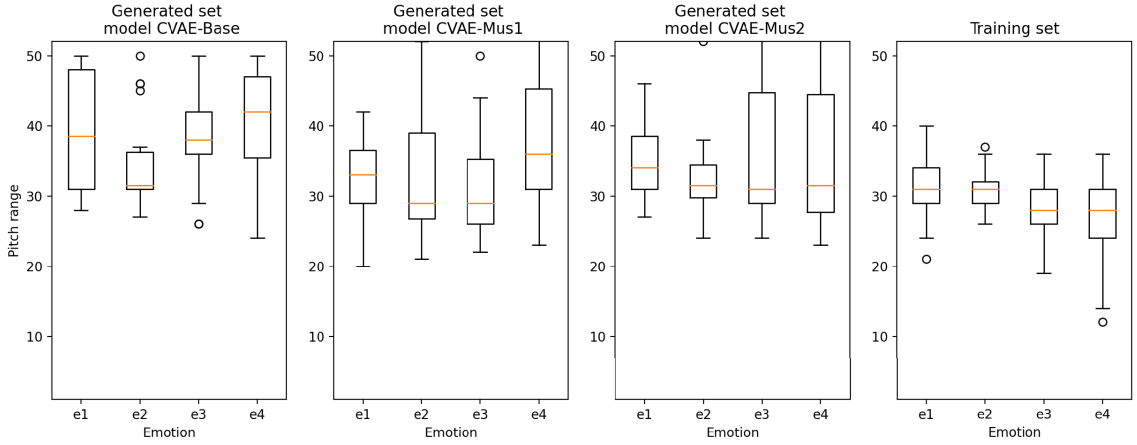

FIGURE 11. ***Pitch range*** **for the generated and training sets labeled with emotions.**

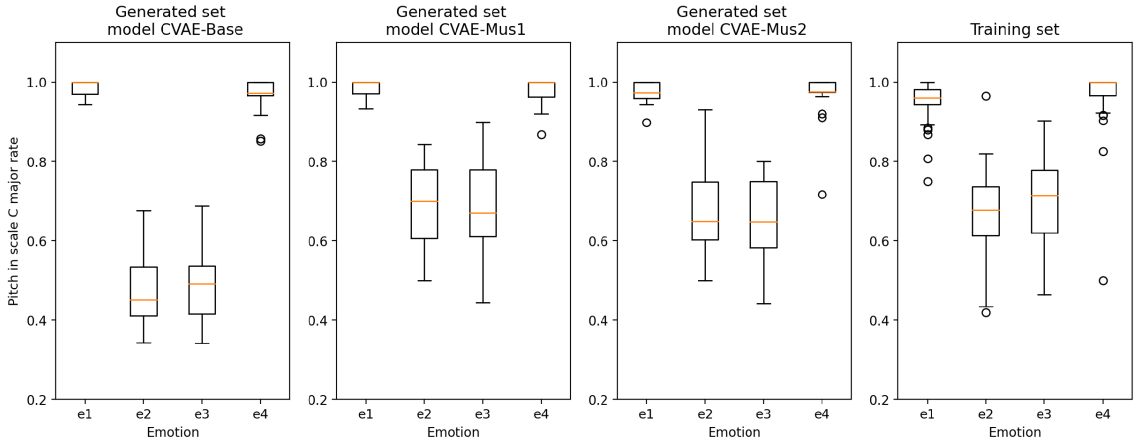

FIGURE 12. ***Pitch in scale C major rate*** **for the generated and training sets labeled with emotions.**

and the proposed model (CVAE-Mus2). The task of each music expert was to listen and determine the emotions for all the examples generated by a given model, i.e. making 80 annotations for the evaluated model. The annotated exam-

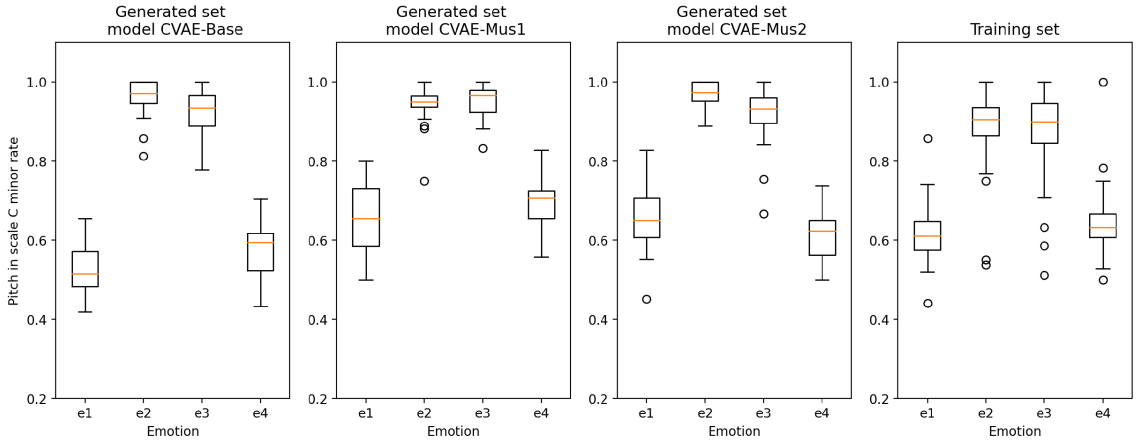

FIGURE 13. ***Pitch in scale C minor rate*** **for the generated and training sets labeled with emotions.**

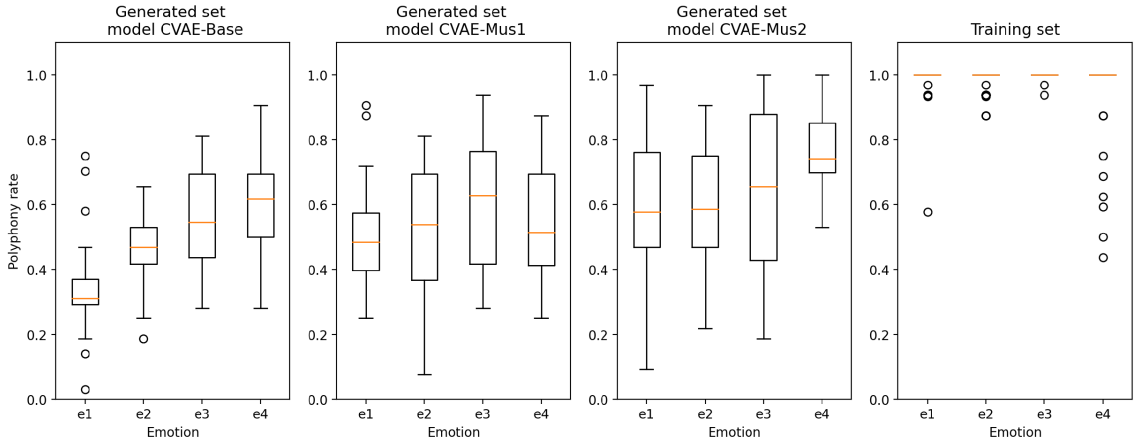

FIGURE 14. ***Polyphony rate*** **for the generated and training sets labeled with emotions.**

ples were mixed up so that their order was not grouped by emotion or model. The task for any music expert was quite tedious as it involved evaluating 240 files of 8 s. each.

Additionally, each music experts was asked to rate on a five-point Likert scale such parameters as:

- *humanness* - whether it sounds like music made by humans;
- *richness* - whether the musical content is interesting.

The obtained annotations from the music experts were averaged.

Expert emotion annotations of the generated set by the baseline model (CVAE-Base), intermediate model (CVAE-Mus1), and by the proposed model (CVAE-Mus2) are presented in Table 4. The values in the rows refer to the generated files with a given emotion, and the values in the columns to files with a specific emotion determined by experts.

Comparing the accuracy of the three tested models, the CVAE-Mus2 model achieved the highest accuracy 68%, compared to the other two: CVAE-Mus1 - 51%; and CVAE-Base - 48% (Table 4). Models CVAE-Mus1 and CVAE-Base have difficulty generating files with emotions e3 and e4. It can be concluded that the weaker models have a problem with generating files with low arousal. The proposed model CVAE-Mus2 in most cases generates a file with the correct emotion, and possible mistakes occur on the arousal axis, i.e. between emotions e1 and e4, and between e2 and e3.

The evaluation using emotions annotated by music experts is consistent with the obtained metrics (Section VI-A), where the tested models were dealt with using the notes of the C major and C minor scales to generate positive and negative emotions (on the valence axis). It can be concluded that limiting the training data to two scales, major and minor, positively influenced the generation of files on the valence axis. Usually, generative models have trouble generating files that differ in valence [27], [28], but in our case through special data preparation (transposition to C major or C minor) this problem was reduced.

Expert annotations of the generated sets evaluated by *humanness* and *richness* are presented in Table 5. A higher

TABLE 4. Expert emotion annotations of the generated set by CVAE-Base, CVAE-Mus1 and CVAE-Mus2 model.

		Expert opinions											
		CVAE-Base				CVAE-Mus1				CVAE-Mus2			
		e1	e2	e3	e4	e1	e2	e3	e4	e1	e2	e3	e4
Generated set	e1	**12**	5	0	3	**12**	2	1	5	**14**	0	0	6
	e2	3	**12**	3	2	2	**13**	5	0	1	**13**	6	0
	e3	1	**11**	4	4	1	**11**	8	0	0	5	**15**	0
	e4	8	1	0	**11**	**9**	1	2	8	5	3	0	**12**

TABLE 5. *Humanness* and *richness* of the generated sets.

Model	*Humanness*	*Richness*
CVAE-Base	1.52	1.38
CVAE-Mus1	2.71	2.97
CVAE-Mus2	**3.05**	**3.21**

value for CVAE-Mus2 confirms that the proposed model generates much more interesting baseline files than the intermediate model. It is also interesting that the *richness* values exceed *humanness*, which proves that despite the generated music, it does not always resemble the music created by humans, but it is intriguing for the listener.

It can be concluded that when training convolutional networks to teach musical representation, it is worth using special shapes for the convolutional layers that improve the analysis of the visual representation of music. The model obtained in this way (CVAE-Mus2) is better suited for generating music with a specific emotion, which was shown by comparing the metrics of the generated and training sets, as well as expert opinions on the generated examples regarding the perceived emotions, *humanness*, and *richness*.

VII. CONCLUSION

This article presents the process of building a model generating polyphonic music sequences with a selected emotion. A database of training examples labeled with emotions by music experts was created and models based on conditional variational autoencoder were built. A special structure for the convolutional layers in CVAE encoder and decoder was proposed for encoding and decoding visual representations of music examples. The presented two parallel convolutional branches for analyzing polyphonic music examples showed an advantage over the sequential structure of standard convolutional layers.

Evaluation of the generated music examples showed that the sequences obtained using the proposed model are closer to the training set examples than the sequences generated using the baseline and the intermediate models. The evaluation using expert opinions showed a higher accuracy of the proposed model in relation to the others regarding the content of a specific perceived emotion in the generated examples. The proposed model turned out to be the winner also when evaluating the created sequences in terms of musical quality.

The limitations of the presented solution are certainly the short length of the generated sequences. Also, more training examples associated with the costly annotation process could improve the obtained results.

In the future, other variants of the connection structure of the convolutional layers could be explored to study encoded representations of polyphonic music. Also, the implementation of mechanisms for generating longer polyphonic sequences, would increase the practical application of the system. The use of a fuzzy emotion model or emotion descriptions using continuous values would be a continuation of this work. The presented approach does not completely solve the problem, in fact, it indicates directions for further research on generating emotional polyphonic music using generative models.

REFERENCES

[1] K. Daher, D. Saad, E. Mugellini, D. Lalanne, and O. A. Khaled, "Empathic and empathetic systematic review to standardize the development of reliable and sustainable empathic systems," *Sensors*, vol. 22, no. 8, Apr. 2022.
[2] J.-P. Briot, "From artificial neural networks to deep learning for music generation: History, concepts and trends," *Neural Comput. Appl.*, vol. 33, no. 1, pp. 39–65, Jan. 2021.
[3] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," 2020, *arXiv:2011.06801*.
[4] H.-W. Dong, W.-Y. Hsiao, and Y.-H. Yang, "Pypianoroll: Open source Python package for handling multitrack pianoroll," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2018, pp. 101–108.
[5] C. C. Pratt, "Music as the language of emotion," Washington, U.S. Govt. Print. Off., Library Congr., Washington, DC, USA, 1950.
[6] S. Ji, X. Yang, and J. Luo, "A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges," *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1–39, Jan. 2024.
[7] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 1–30, Sep. 2017.
[8] Z. Zhao, H. Liu, S. Li, J. Pang, M. Zhang, Y. Qin, L. Wang, and Q. Wu, "A review of intelligent music generation systems," 2022, *arXiv:2211.09124*.
[9] L. Yang, S. Chou, and Y. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. 18th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2017, pp. 324–331.
[10] C. A. Huang, T. Cooijmans, A. Roberts, A. C. Courville, and D. Eck, "Counterpoint by convolution," in *Proc. 18th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2017, pp. 211–218.
[11] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "MusicLM: Generating music from text," 2023, *arXiv:2301.11325*.
[12] Y.-H. Yang and H. Chen, "Machine recognition of music emotion: A review," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 3, pp. 1–30, May 2012.
[13] L. Lu, D. Liu, and H.-J. Zhang, "Automatic mood detection and tracking of music audio signals," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 1, pp. 5–18, Jan. 2006.
[14] J. Grekow, "Audio features dedicated to the detection of four basic emotions," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manag.*, Warsaw, Poland, Sep. 2015, pp. 583–591.

[15] B. G. Patra, D. Das, and S. Bandyopadhyay, "Labeling data and developing supervised framework for Hindi music mood analysis," *J. Intell. Inf. Syst.*, vol. 48, no. 3, pp. 633–651, Jun. 2017.
[16] J. A. Russell, "A circumplex model of affect," *J. Personality Social Psychol.*, vol. 39, no. 6, pp. 1161–1178, Dec. 1980.
[17] F. Weninger, F. Eyben, and B. Schuller, "On-line continuous-time music mood regression with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5412–5416.
[18] J. Grekow, "Music emotion recognition using recurrent neural networks and pretrained models," *J. Intell. Inf. Syst.*, vol. 57, no. 3, pp. 531–546, Aug. 2021.
[19] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, "Music mood detection based on audio and lyrics with deep neural net," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Sep. 2018, pp. 370–375.
[20] J. Grekow, *From Content-Based Music Emotion Recognition to Emotion Maps of Musical Pieces* (Studies in Computational Intelligence), vol. 747. Cham, Switzerland: Springer, 2018.
[21] R. Panda, R. Malheiro, and R. P. Paiva, "Audio features for music emotion recognition: A survey," *IEEE Trans. Affect. Comput.*, vol. 14, no. 1, pp. 68–88, Jan. 2023.
[22] L. Turchet and J. Pauwels, "Music emotion recognition: Intention of composers-performers versus perception of musicians, non-musicians, and listening machines," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 305–316, 2022.
[23] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," *Exp. Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118190.
[24] D. Williams, A. Kirke, E. R. Miranda, E. Roesch, I. Daly, and S. Nasuto, "Investigating affect in algorithmic composition systems," *Psychol. Music*, vol. 43, no. 6, pp. 831–854, Nov. 2015.
[25] L. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2019, pp. 384–390.
[26] R. Madhok, S. Goel, and S. Garg, "SentiMozart: Music generation based on emotions," in *Proc. 10th Int. Conf. Agents Artif. Intell.*, 2018, pp. 501–506.
[27] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on LSTM networks," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Automat. Control Conf.*, 2019, pp. 2039–2043.
[28] H. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proc. 22nd Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2021, pp. 318–325.
[29] S. Sulun, M. E. P. Davies, and P. Viana, "Symbolic music generation conditioned on continuous-valued emotions," *IEEE Access*, vol. 10, pp. 44617–44626, 2022.
[30] M. A. Pangestu and S. Suyanto, "Generating music with emotion using transformer," in *Proc. Int. Conf. Comput. Sci. Eng. (ICSE)*, vol. 1, Nov. 2021, pp. 1–6.
[31] P. L. T. Neves, J. Fornari, and J. B. Florindo, "Generating music with sentiment using transformer-GANs," in *Proc. 23rd Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2022, pp. 717–725.
[32] Y. Zhang, "Representation learning for controllable music generation: A survey," in *Proc. 21st Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2020, pp. 1–8.
[33] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. 35th Int. Conf. Mach. Learn. (PMLR)*, 2019, pp. 4364–4373.
[34] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proc. 21st Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2020, pp. 662–669.
[35] R. Guo, I. Simpson, T. Magnusson, C. Kiefer, and D. Herremans, "A variational autoencoder for music generation controlled by tonal tension," in *Proc. Joint Conf. AI Music Creativ.*, Oct. 2020, pp. 1–12.
[36] M. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2010, pp. 637–642.
[37] *List of Works in the Music21 Corpus*. Accessed: Jun. 9, 2023. [Online]. Available: https://web.mit.edu/music21/doc/about/referenceCorpus.html
[38] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for Bach chorales generation," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1362–1371.
[39] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1–8.
[40] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1881–1888.
[41] J. Grekow and T. Dimitrova-Grekow, "Monophonic music generation with a given emotion using conditional variational autoencoder," *IEEE Access*, vol. 9, pp. 129088–129101, 2021.
[42] A. Gabrielsson, "Emotion perceived and emotion felt: Same or different?" *Musicae Scientiae*, vol. 5, pp. 123–147, Sep. 2001.
[43] A. Aljanaki, Y. H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLoS ONE*, vol. 12, no. 3, pp. 1–22, 2017.
[44] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A toolkit for symbolic music generation," in *Proc. 21st Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2020, pp. 1–8.
[45] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
[46] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

JACEK GREKOW received the M.S. degree in computer systems from the Technical University of Sofia, Bulgaria, in 1994, the B.S. degree in music from Vienna Conservatoire (Konservatorium der Stadt Wien), Austria, in 1996, the M.S. degree in music from the Department of Instrumental and Educational Studies, Fryderyk Chopin University of Music, Warsaw, Poland, in 2006, and the Ph.D. degree in computer science from the Faculty of Information Technology, Polish-Japanese Academy of Information Technology, Warsaw, in 2009. He is currently an Associate Professor with the Faculty of Computer Science, Bialystok University of Technology, Poland. His research interests include data mining, music emotion recognition, music generation, and deep learning.