POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

# Procedural music generation for video games conditioned through video emotion recognition

Laurea Magistrale in Music and Acoustic Engineering

**Author:** Francesco Zumerle

**Advisor:** Prof. Massimiliano Zanoni

**Co-advisor:** Luca Comanducci

**Academic year:** 2022-2023

## 1. Introduction

Music plays a crucial role in the world of video games, and its relationship with technology and computer science has always been closely intertwined. Nowadays, with the recent advancements in deep learning, there are exciting opportunities for reimagining how music is composed for this medium. Specifically, open-world video games, characterized by non-linear storytelling and a multitude of gameplay scenarios offered, have become incredibly popular. Crafting a soundtrack that can adapt to a wide range of in-game events and variations presents a significant challenge, as it's exceptionally difficult for human composers to anticipate every possible situation. Furthermore, the recent success of a few indie games has highlighted a growing interest among both developers and players in creating artistic and emotionally engaging experiences that feature high levels of musical and visual interactivity. In this context, in this thesis's summary we propose a system that constantly analyzes a game's video stream, predicts the emotions elicited in the player during each set of video frames and continuously generates music that aligns with those emotions [5]. To the best of our knowledge, this is the first research proposing and validating a framework of this type. For the emotion detection task, we trained a 3D CNN with the LIRIS-ACCEDE dataset [1], composed of short movie excerpts associated with Valence and Arousal values. Then, we employed a pre-trained Music Transformer [3] able to generate symbolic music conditioned on the same two continuous values, customizing its inference algorithm so it generates musical continuation starting from any input MIDI. Next, combining these two blocks we built the desired architecture. Finally, we assess our model's effectiveness by having human participants play a real video game, watch videos with emotion-based and non-emotion-based music, and rate them through an emotion annotation task and a questionnaire.

The contents of this summary are organized as follows. In *Section 2* we formally define our proposed system. *Section 3* describes the two blocks composing our framework: video emotion detection and conditioned music generation. In *Section 4*, after evaluating the proposed 3D-CNN for emotion detection, we present the perceptual test performed, analyzing and discussing the results collected. Lastly, *Section 5* is devoted to conclusions and future developments.

## 2. Problem Formulation

Formally, we define a single video frame as

$$f = \left[ \begin{bmatrix} a_{1,1} & \dots & a_{1,w} \\ \dots & \ddots & \dots \\ a_{h,1} & \dots & a_{h,w} \end{bmatrix}_R \begin{bmatrix} \ddots \end{bmatrix}_G \begin{bmatrix} \ddots \end{bmatrix}_B \right], \tag{1}$$

where $h, w \in \mathbb{N}$ represent the height and width of the frame, $R, G, B$ are the three channels corresponding to the RGB color model, and $a \in [0, 1] \subset \mathbb{R}$ indicates the value of each pixel for each color channel. Hence, we can describe our system devoted to the emotion detection task as a function $\mathcal{S}_1$ that, given as input a sequence of video frames $\mathbf{f} = [f_1, \dots, f_N]$ returns two continuous values $V$ and $A$, representing respectively the predicted Valence and Arousal. Consequently, this first component can be modeled as

$$[V, A] = \mathcal{S}_1(\mathbf{f}), \tag{2}$$

where $V \in [-1, 1] \subset \mathbb{R}$ and $A \in [-1, 1] \subset \mathbb{R}$.
Moving to the conditioned music generation task, let $\mathbf{m}$ be a sequence of music tokens $\mathbf{m} = [T_1, \dots, T_N]$, where each music token $T$ is a tuple of two elements $T = <i, v>$, with $i$ being an index corresponding to a midi event, such as DRUMS_OFF or TIMESHIFT, while $v$ determines its correspondent value (e.g. pitch for note or time for timeshift). Formally, we want to build a function $\mathcal{S}_2$ defined as

$$\mathbf{m}_{out} = \mathcal{S}_2([V, A], \mathbf{m}_{primer}), \tag{3}$$

where $V, A$ are the conditioning Valence and Arousal values already defined for equation (2), while $\mathbf{m}_{primer}$, $\mathbf{m}_{out}$ are two sequences of music tokens, that correspond respectively to the input conditioning MIDI file and the output MIDI file. These two distinct architectures can be easily combined by setting the predicted values of equation (2) as the input values of equation (3), obtaining

$$\mathbf{m}_{out} = \mathcal{S}_3(\mathbf{f}, \mathbf{m}_{primer}), \tag{4}$$

which is the complete proposed architecture, also illustrated in Figure 1. As a result, we defined a system that processes progressively a consecutive fixed number of input video frames, obtains two continuous values representing the emotion elicited by the video and generates music based on this information.

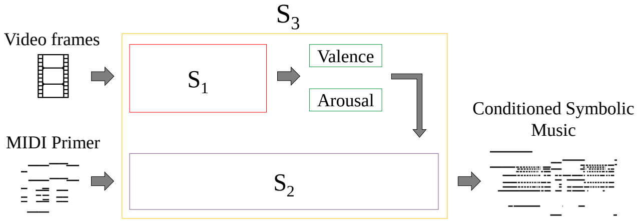

Figure 1: Scheme of the proposed system, as defined in section 2

## 3. Proposed Method

In this section we first analyze our proposed CNN for emotion prediction (subsection 3.1). Afterwards, we briefly present the pre-trained music transformer employed for conditioned music generation (subsection 3.2).

### 3.1. Video emotion detection

Regarding our dataset, we initially process each movie clip extracting a fixed number of frames and resizing them to a desired resolution. Moving to the valence-arousal labels associated to every video, for each affective dimension we first perform a min-max normalization along the series of all labels, and then rescale the obtained values in the interval $[-1, 1]$.
We propose a 3D-CNN based on the ResNet network with 18 layers. Specifically, we replace each convolutional layer with a (2+1)D convolutional layer proposed in [4], composed of a spatial 2D convolution followed by a temporal 1D convolution, which is equivalent to a 3D convolution. Compared to a classical 3D layer, this method increases the number of nonlinearities and simplifies the optimization process. As a result, each residual block is composed of two (2+1)D convolutional layers with a ReLU between them.
In detail, our proposed network depicted in Figure 2 is composed of:

- A (2+1D) convolutional layer (with 16 filters and kernel size=$(3, 7, 7)$), followed by Batch Normalization, ReLU, a downsampling layer and dropout with rate 0.3.
- A Residual Block (16 filters and kernel size=$(3, 3, 3)$), followed by a downsampling layer and dropout with rate 0.4.
- Another Residual Block (32 filters and kernel size=$(3, 3, 3)$), followed by a downsampling layer and dropout with rate 0.3.
- Another Residual Block (64 filters and kernel size=$(3, 3, 3)$), followed by a downsam-

pling layer and dropout with rate 0.4.
- A last Residual Block (128 filters and kernel size=$(3,3,3)$), followed by a global average pooling and a dense layer with 2 neurons.

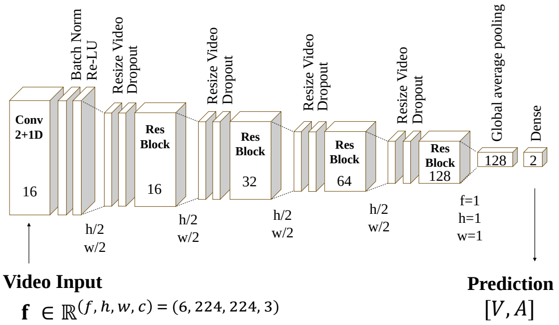

Figure 2: Scheme of the proposed 3D-CNN. The last dimension, initially representing the *RGB* channels, indicates the feature maps obtained after each conv layer. Although the input is $4D$, we display $3D$ blocks since the first dimension $f$ remains unchanged until global average pooling.

### 3.2. Music generation conditioned on emotions

For generating emotionally conditioned music, we modify a pre-trained music transformer proposed by Sulun et al. [3] in order to compose musical continuation of a desired input MIDI. In practice, before each generation, a MIDI file denoted as *primer* is converted to a sequence of music tokens $\mathbf{m}$, as defined in equation (3), and fed to the transformer at the beginning of the generation process. Regarding the transformer architecture, we selected the variant named *continuous-concatenated*, that showed the best performance in generating symbolic music conditioned on continuous valence-arousal values.

## 4. Experimental Evaluation

This section begins by presenting the 3D-CNN training procedure (Sec. 4.1). Then, the complete implementation of our approach is explained (Sec. 4.2). Finally, we describe the subjective evaluation procedure (Sec. 4.3), analyzing (Sec. 4.4) and discussing (Sec. 4.5) the obtained results [1].

### 4.1. 3D-CNN Training procedure

We trained our 3D-CNN with a subset of the LIRIS-ACCEDE dataset consisting of 8000 movie excerpts and we split them into Train, Validation and Test sets, with a dataset split policy of 80-10-10. For each video we extract 6 frames and resize them to width $= 224$, height $= 224$. While training for a maximum of 1000 epochs, we monitor validation loss using Mean Squared Error, employing an early stopping policy after 100 epochs. After being trained, our model reached its best performance at epoch 301, with a MSE of 0.291 for Train, 0.338 for Validation and 0.337 for Test sets.

### 4.2. Complete pipeline implementation

We set our 3D-CNN so that for any input video it predicts a time series for both Valence and Arousal, extracting 6 frames every 1 second and returning the correspondent values. With this information, we designed an algorithm able to analyze the emotion evolution over time, determining at each time step whether to generate music based current emotion or keep the previous conditioning. Formally, let $\mathbf{x} = [x_1, \dots, x_N]$ be a sequence of length $N$ containing the annotations of a single affective dimension, computed for each second $s$ of current video. For both Valence and Arousal, we calculate the *Absolute value of the Difference Quotient* $Q^{abs}$ (which represents how abruptly or smoothly a quantity changes) between each couple of subsequent predictions $(x_i, x_{i-1})$, defined as

$$Q^{abs}(\mathbf{x}) = \begin{cases} \dfrac{|x_i - x_{i-1}|}{2}, & \forall i : 1 < i \leq N, \\ 0 & , \forall i : i = 1. \end{cases} \quad (5)$$

As a result, we obtain a sequence $\mathbf{q} = [Q^{abs}(x_1), \dots, Q^{abs}(x_N)]$ for each affective time series. Next, the resulting sequences are used to simulate the real-time behaviour of our model, generating a MIDI soundtrack that evolves coherently according to each video. Precisely, we designed Algorithm 1 as a function $\mathcal{C}$ that receives as input the predicted sequences of valence and arousal $\mathbf{v}, \mathbf{a}$, the sequences of absolute value of difference quotients $\mathbf{q}_{\text{val}}, \mathbf{q}_{\text{aro}}$, two parameters $p$ (percentile threshold) and $d_{\text{th}}$ (minimum duration threshold), and generates a sequence $\mathbf{m}$ of music tokens, which can be modeled as

$$\mathbf{m} = \mathcal{C}(\mathbf{v}, \mathbf{a}, \mathbf{q}_{\text{val}}, \mathbf{q}_{\text{aro}}, p, d_{\text{th}}), \quad (6)$$

[1] all code available from this Github link.

with $p \in [0, 100] \subset \mathbb{N}$ and $d_{\text{th}} \in \mathbb{N}$. During the generation process we want to change emotional conditioning only when most abrupt emotional variations occur. To do so, we compute the $p$-th percentile for both $\mathbf{q}_{\text{val}}$ and $\mathbf{q}_{\text{aro}}$, resulting in two values, $v_{\text{th}}$ and $a_{\text{th}}$. These values are compared each second with the current $Q^{abs}$ of the corresponding emotion. If the previous generation has already exceeded a length of $d_{\text{th}}$ and one of the current quotients exceeds its threshold, then the generation process restarts with new values.

Algorithm 1 Conditioned music generation

```
Input: $\mathbf{v}$, $\mathbf{a}$, $\mathbf{q}_{\text{val}}$, $\mathbf{q}_{\text{aro}}$, $p$, $d_{\text{th}}$
$v_{\text{th}} \leftarrow$ compute $p$-th percentile of $\mathbf{q}_{\text{val}}$
$a_{\text{th}} \leftarrow$ compute $p$-th percentile of $\mathbf{q}_{\text{aro}}$
$d_{\text{curr}} \leftarrow 0$
$\mathbf{m} \leftarrow []$
begin new music generation conditioned on current $\mathbf{v}[s], \mathbf{a}[s]$ values
$s \leftarrow 0$
for $s <$ duration in seconds of $\mathbf{v}$ do
    if $d_{\text{curr}} \geq d_{\text{th}}$ and $(\mathbf{q}_{\text{val}}[s] > v_{\text{th}}$ or $\mathbf{q}_{\text{aro}}[s] > a_{\text{th}})$ then
        $\mathbf{m}.append$(current music generation)
        begin new music generation conditioned on current $\mathbf{v}[s], \mathbf{a}[s]$ values
        $d_{\text{curr}} = 0$
    else
        continue previous music generation
        $d_{\text{curr}} = d_{\text{curr}} + 1$
    end if
    $s \leftarrow s + 1$
end for
Output: $\mathbf{m}$ containing the final soundtrack
```

For our experiments, we chose $p = 80$, $d_{\text{th}} = 3$, meaning that we generate music changing emotional conditioning only in top-20% biggest emotional variations and only when the previous conditioning has already produced at least 3 seconds of music.

### 4.3. Subjective evaluation setup

To test the effectiveness of our complete proposed method, we applied it to *No Man's Sky*, an action-adventure survival game, set in an open-world galaxy and characterized by a first person perspective, which according to our preliminary tests provides the ideal visual feedback for the emotion prediction task. In particular, we designed a perceptual test following the procedure described in [2], adapting it to our specific case. First of all, we recorded gameplay videos of *No Man's Sky* and we extracted 40 videos of 30$s$ duration each, specifically selected for containing clear visual or emotional changes. These clips were subsequently divided in 4 groups of 10, each one with original sound effects and the following music categories: **Conditioned**, generative music conditioned on valence-arousal; **Unconditioned**: unconditioned generative music; **Original**: original soundtrack; **None**: no music. Once the material was prepared, we designed a blind and randomized procedure for evaluating the effectiveness of our framework, combining an affective annotation task and a comparative questionnaire. Both these tasks were performed using one video for each category, for a total of 4 showed to each participant, selected from the complete corpus of 40 clips.

A single session articulates in 3 phases:

*Gameplay session.* The participant is given 15 minutes to play *No Man's Sky*, familiarizing with its commands and game flow. A fixed list of instructions is given to the subject, making sure that he experiences most relevant gameplay components. During the playthrough the music is muted, in order to prevent biases in subsequent evaluations.

*Emotion annotation task.* The participant is presented a brief explanation of a single affective dimension and is then asked to watch and annotate in real-time all 4 videos currently assigned, depending on how he feels according to the described emotion. The process repeats for both affective dimensions. Valence was presented as the emotion that "Indicates how happy or sad (unhappy) you feel", while Arousal "Indicates how excited or calm you are feeling". The Self-Assessment Manikin was used to better clarify these concepts. This task was implemented following a state-of-the-art approach named Rank-Trace: while a video is being played, the user presses the keyboard arrows up or down every time he feels an increase or decrease of the chosen emotion. Button inputs are received unbounded every 200 ms and a live plot gives a visual feedback of the annotation process so far. We opted for 30 seconds clip duration in order to keep the annotation task relatively short, preventing fatigue that would have caused poor an-

notations.
*Questionnaire.* Lastly, the participant opens a questionnaire where the same 4 videos used in previous task can be reproduced again, if needed. They are asked to pick one video for each of the following questions:

1. In which video do you feel the music most closely matches the events and actions of the gameplay? (gameplay match)
2. In which video do you feel that the music most closely matches the emotion that you perceive from the gameplay? (emotion match)
3. In which video did you feel most immersed in the gameplay? (immersion)
4. Which video's music did you enjoy the most? (preference)

Each question will be referenced in future discussions with the words between parenthesis. After that, information regarding anagraphic and gaming experience, as well as optional comments are collected.

## 4.4. Results

34 subjects ranging from 18 to 29 years old, average age = 23.8, participated to the subjective evaluation. Among these, 20 use he/him pronouns, 13 identify as she/her and one chose they/them. 67.6% of the participants spend less than four hours per week playing video games, while the remaining 32.4% spend more than 4 hours weekly. Each experimental session lasted 30 minutes on average.

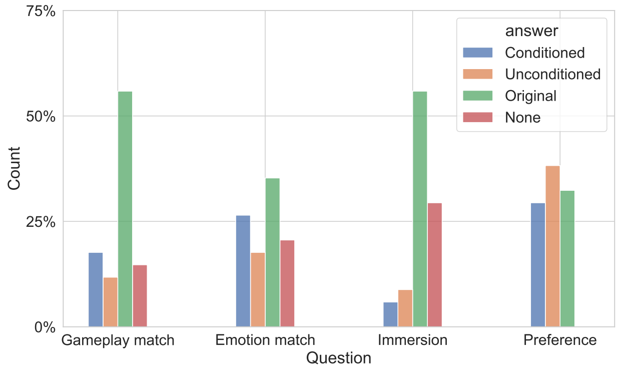

Figure 3: Questionnaire results

All the collected time series were compared to our model's predictions according to Distance, performed with Dynamic Time Warping (DTW), and the Root mean squared error (RMSE). The DTW Distance allows for a flexible alignment of the two sequences, even when they have different lengths or temporal distortions. Before performing the calculations, all values of both annotations and predictions were normalized with z-score normalization, computed globally across valence and arousal. Afterwards, for each user's annotation we calculated its metrics by comparing it with the model's predictions for the same video. The obtained results are summarized in Table 1, where the average value is computed across each of the four video categories, first separating the affective dimensions, then merging them. For each average metric computed, we additionally display its Standartd Error of Mean (SEM), a statistical measure that quantifies the value's variability as an estimate of the population mean.

Affective Dimension: Valence

| Measure | Cond | Uncond | Original | None |
|---|---|---|---|---|
| Distance | 15.164 | 16.741 | 22.006 | 23.529 |
| SEM | 1.221 | 2.074 | 1.674 | 2.313 |
| RMSE | 1.006 | 1.113 | 1.37 | 1.460 |
| SEM | 0.070 | 0.129 | 0.096 | 0.130 |

Affective Dimension: Arousal

| Measure | Cond | Uncond | Original | None |
|---|---|---|---|---|
| Distance | 14.985 | 16.737 | 18.129 | 15.662 |
| SEM | 1.288 | 1.337 | 1.917 | 1.509 |
| RMSE | 1.061 | 1.127 | 1.248 | 1.084 |
| SEM | 0.085 | 0.081 | 0.13 | 0.088 |

All Affective Dimensions

| Measure | Cond | Uncond | Original | None |
|---|---|---|---|---|
| Distance | 15.074 | 16.739 | 20.067 | 19.476 |
| SEM | 0.881 | 1.225 | 1.285 | 1.439 |
| RMSE | 1.033 | 1.120 | 1.309 | 1.266 |
| SEM | 0.055 | 0.075 | 0.081 | 0.080 |

Table 1: Annotation task Results

For assessing validity of our measurements, we tested the assumption of normality using D'Agostino-Pearson omnibus test: below we will discuss only results for which the test was passed. Regarding Valence dimension, we performed ANOVA test comparing each categoriy's variance for both metrics, obtaining $p$-value $<$ 0.05 for Distance. Then, a post hoc Tukey test showed that Conditioned music leads to significantly better results than Original soundtrack and No music, while other comparisons were not statistically relevant. After merging the affective dimensions, ANOVA was passed for both Distance and RMSE. The post hoc test confirmed that videos with Conditioned music had lower errors with respect to the Original ones. Lastly, merging all categories and comparing Valence and Arousal average metrics, ANOVA and post hoc tests determined a significantly better performance of the latter, for Distance metric.

Regarding the questionnaire results, Figure 3 shows the answer distribution for all questions.

### 4.5. Discussion

Regarding the emotion annotation results, statistical tests suggest that our model effectively elicits the desired affective evolution, significantly improving similarity between annotations and predictions with respect to original soundtrack. The comparison between Conditioned and Unconditioned annotations yields inconclusive results, although both RMSE and Distance our model's predictions are always more similar to the first category. Emotions are highly subjective and it's not surprising to see the highest variance (SEM in Table 1) among annotations performed without music, suggesting that multiple users can perceive completely different emotions while watching the same video. Let's now move to the questionnaire results. Comparing Conditioned and Unconditioned music, the majority of voters found music of the first category more coherent to both gameplay events and their perceived emotions, confirming the effectiveness of our approach. Then, we observe that for the first three questions original soundtrack outperforms our generative approach, while for the question regarding musical "Preference" the situation is the opposite. This suggests that while the music transformer produces music of high quality, its application in the video game context is still challenging and requires further improvements. Among all, an affective dataset of "ambient" music, the genre of the original soundtrack, would for sure have improved our results, although currently it does not exist.

## 5. Conclusions

In this work we presented a complete pipeline for generating a continuous video game soundtrack emotionally coherent with gameplay video. The results of our experiments confirm the validity of our method, while providing useful insight for future researches in this field. Currently, the lack of video game affective datasets with valence-arousal annotation constitutes a critical limit for emotion prediction performance. Regarding affective music datasets, current options cover a low variety of genres, preventing versatility and limiting musical coherence with most video games. Lastly, the real-time implementation of our system remains a future work, although the recent diffusion of cloud gaming provides a concrete solution for the computationally demanding inference of DL models. While not definitive, these results show that emotion-conditioned procedural music generation for video games is a path pursuable, that could enrich and bring innovation in the context of open-world games and non-linear experiences.

## References

[1] Yoann Baveye, Emmanuel Dellandrea, Christel Chamaret, and Liming Chen. Liris-accede: A video database for affective content analysis. *IEEE Transactions on Affective Computing*, 6(1):43–55, 2015.

[2] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Tchemeube. Preglam-mmm: Application and evaluation of affective adaptive generative music in video games. In *Proceedings of the 17th International Conference on the Foundations of Digital Games*, pages 1–11, 2022.

[3] Serkan Sulun, Matthew EP Davies, and Paula Viana. Symbolic music generation conditioned on continuous-valued emotions. *IEEE Access*, 10:44617–44626, 2022.

[4] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[5] Francesco Zumerle, Luca Comanducci, Massimiliano Zanoni, Alberto Bernardini, Fabio Antonacci, and Augusto Sarti. Procedural music generation for videogames conditioned through video emotion recognition. (accepted to the 4th International Symposium on the Internet of Sounds - IS$^2$ 2023).