CONSERVATORIO CESARE POLLINI DI PADOVA

DIPARTIMENTO DI NUOVE TECNOLOGIE E LINGUAGGI MUSICALI

Diploma accademico di 2° livello in Tecniche Informatiche di Analisi e Valorizzazione dei Materiali Musicali

A Neural Network Based Framework for Archetypical Sound Analysis and Re-Synthesis

Diplomando: Eric Guizzo

Matricola: 12361

Relatore: M° Nicola Bernardini

Correlatore: Ing. Antonio Rodà

Anno accademico: 2017-2018

TABLE OF CONTENTS

1 Abstract

This research describes a preliminary approach to algorithmically reproduce the archetypical structure adopted by humans to imagine and classify sounds. This involves the identification of the processes through which our intellect constructs mental representations of experienced timbres, defining perceptual categories of audio information. Humans are able to recognize and imagine sound events through a psychic elaboration that is intrinsically related to these categorical representations. Therefore, a numerical model of the latter would provide the possibility of computationally manage audio information in a similar manner as the human brain does. Such investigation would provide a solid starting point to develop a novel method for audio analysis, generation and manipulation, aimed at assisting artists and musicians in creative contexts. This approach would confer a "natural" character to sound synthesis and processing by offering the possibility of modeling timbres in a spontaneous manner, directly reflecting our inner process of conceiving sounds. Moreover, this would permit an artist to bypass the necessity of a consolidated knowledge in the fields of acoustics and signal processing, by allowing to administrate perceptual-oriented parameters of sound. The chosen approach is based on the implementation of an artificial intelligence algorithm, specifically designed to perform automatic signal classification and data synthesis tasks. To reach this target, consolidated techniques of digital audio processing and deep learning have been connected in a single framework. In this instance, we focused on modeling the human perception of *chaos* and *order* in sound information, with the prospect of applying the developed strategies also for other abstract/perceptual sound features. The obtained framework is capable of predicting the human perceived chaos/order level of audio signals, as well as synthesizing timbres that present a desired amount of this feature. Furthermore, this method has been be applied in a

practical situation, in order to demonstrate its expressive potentialities in a performance-oriented circumstance. We finally discuss the obtained accuracy and possible implications in disparate contexts.

2 INTRODUCTION

The technological developments in recent years are leading to increasingly valorize the computing paradigm of *learning algorithm,* around which is structured the concept of Artificial Intelligence (AI). By imitating the biological functioning of the human mind, the "silicon brain" is able to store and elaborate any kind of experience, learning from it how to conduct disparate tasks, rivaling, and even overtaking human performance. On the one hand, this led to important innovations in the scientific field. Consider, for example, the possibility of performing automatic diagnosis of medical diseases, automatically driving means of transport or computationally interpreting human language. From another point of view, the AI provided to artists a new aesthetic and conceptual dimension to investigate. In particular, the interaction between the human being and the electronic surrogate of himself constitutes a fertile and suggestive breeding ground to be explored, representing which is probably the most emblematic ethic, social and scientific theme of our time.

The research project treated in this thesis fits in this ideological context, investigating from a technical, practical and artistic point of view a specific and restricted utilization instance of the AI for musical scopes. The stimulating and fascinating challenge of creating an algorithmic replica of human sound perception led us to the definition of the *archetypical* sound analysis and re-synthesis model. The path proposed in this work aims at illustrating and justifying the implementation of this algorithm, describing the technical and theoretical background on which is based and critically analyzing its defects and potentialities. In the core of this thesis we report the 3 principal phases of our research: critical examination of previous literature (background), technical report of the development and implementation of the analysis and re-synthesis models (method), description of an artistic performance entirely realized through the achieved algorithms (application). During the design and development of this

project, countless conceptual and technical complications emerged, which led to several compromises and also sharp changes of direction. In order to present a more fluent and useful essay, we decided to report only the final implementation, exhaustively justifying the decisions undertaken on each occasion. Moreover, for a brief reading, a short summary is provided at the end of every important chapter and sub-chapter.

Be clear from the beginning that the target of this document is the electronic musician. In fact, particular attention has been paid to maintain a discursive format, explaining the treated subjects as clearly and comprehensibly as possible also for non-specialists, notwithstanding the relative complexity of the arguments. Furthermore, we have sought to provide all the necessary connections to deepen the most sophisticated concepts. Thus, this work can be considered also an introductory guide to the use of the AI for musical purposes (as well as a practical implementation example), targeted to persons who have a basic understanding of signal processing for audio applications.

2.1 On sound archetypes

In *Traité des Objets Musicaux* [1], *Pierre Schaeffer* identified an absolute paradigm through which univocally classify sound events, by specifying a selected set of perceptual features evoked by the sounds[1]. These features were supposed to be universal, and then equally perceived by different individuals. Therefore, this criterion allowed the identification of timbre classes, characterized by specific perceptual connotations.

An important corollary of this theory is the concept of *Temporal Semiotic Unity* (TSU), emerged at the *Laboratoire de Musique et d'Informatique de Marseille* in 1991[2]. The TSU is based on the evolution

1 Mass, dynamics, timbre, melodic profile, mass profile, grain and look.

from the concept of *Sound Object* to the idea of a *Semiotic Sound Object*. This implicates a separation from Schaeffer's pure gestalt-oriented view, which considers the timbre as an entity completely isolated from its context, basing on the conception of a "limited listening" that ignores any "causal or associative meaning" of the sound object [2]. Conversely, TSUs take in consideration the *semantic value* of sound material, intended as the multi-sensorial and metaphoric associative field related to auditory information. The concept of TSU can refer both to temporal succession of sound events (i.e. musical figures) and to the evolution (through time) of the timbre structure within a single sound event. Therefore, TSUs can be identified as *semantic auditory archetypes*. These are intended as perceptual categories of sound events (or musical figures) capable of triggering complex mental associations and evoke specific emotions related to their semantic connotation. Nevertheless, they can not be considered as *absolute descriptors,* being based on mental connections that are strictly dependent from the cultural background and the experience of an individual [2]. This is empirically evident, in fact, for instance, a *blue* sound could mean something completely different for distinct persons. On the contrary, it is more probable that diverse individuals would be more concordant in the conception of a *bright* timbre. Thus, it is fundamental to consider the *ambiguity level* of an archetype for its definition. The higher is its ambiguity, the higher is its perception subjectivity. Accordingly, from a theoretical point of view, this research aims to algorithmically model TSUs referred to single sound events (not to musical figures), taking into account their intrinsic ambiguous character.

To precisely connote "the archetypical structure adopted by humans to imagine and classify sounds", we rely on the semiotic tendencies derived by *Lakoff and Johnson*' s *Metaphors We Live By* [3]. In particular, the notion o f *sound archetype* contemplated in our research strongly depends on metaphoric and cross-modal correlations between audio perception and

different semantic/perceptual areas. As largely discussed by the above-mentioned authors, human intellection, imagination, language and interpretation of the sensorial experience are sharply influenced by this type of implications. For instance, within the western culture, it is common to interrelate the concepts of "up" and "positive", and the same with "down" and "negative". This is due to several factors that are common in the background of this culture, such as the fact that a healthy person is usually erected and a diseased one lyes down. Furthermore, following metaphoric correspondences, western people are used to adopt expressions such as "sharp sentence" or "brilliant mind", which can describe a particular character of the notion providing a direct comparison with different concepts (coming from different semantic areas) that are *isomorphically* correlated. The same type of implications occur for the mental classification and elaboration of perceptual stimuli. In fact, humans tend to refer to the latter adopting metaphoric/cross-modal attributes. The latter can be considered as *perceptual archetypes*, intended as *semantic units* able to qualitatively represent the formal appearance of perceptual phenomenons.

This is perfectly compatible with hearing-related intellection. Indeed, it is empirically evident that sound perception is able to evoke allegorical emotions and humans are used to refer to these sensations to describe and categorize audio events. It is common, for instance, the use of terms such as *rounded* to label sounds. Moreover, humans are able to imagine caricatural[2] sounds, materializing these attributes into imaginary audio fluxes. For example, it is plausible to describe the sound produced by an old closing door or a broken celery as *crackly*. These timbres are different, although they share certain physical and perceptual characters that make humans associate them with the same archetype. Moreover, people who experienced

2 A caricatural sound is intended as an imagined sound that do not correspond to the exact memory of a sound event an individual heard, although it is mentally reconstructed by elaborating and combining perceptual characteristics of experienced sounds.

and categorized certain timbres as *crackly* are able to mentally reproduce new sounds matching that specific category, as well as recognizing if a real perceived sound event is or is not a *crackle*. This occurs by analyzing its features and comparing them with the ones of similar experienced timbres, which have been previously extracted and memorized. In fact, as described by *McAdams* [4], human ability of recognition and discrimination of timbre categories suggest a predisposition of encoding spectral and temporal sound properties into isomorphic mental representations, which are capable of imitating and summarizing their appearance. Hence, these representations, externalized through metaphoric/cross-modal attributes, can be considered as the *sound archetypes* that constitute the vocabulary adopted by our imagination to classify audio-related experiences and mentally recall instances of sound categories. It is important to denote that several typologies of metaphoric/cross-modal implications can occur for the definition of a sound archetype [5]. For instance, we could refer to a timbre quality indicating the material that emitted a sound (“metallic”, “wooden”), the source instrument (“violin sound”, “engine sound”), a tactile sensation (“smooth”, “sharp”), a visual sensation (“brilliant”, “dark”) and the list could continue.

By examining the above-mentioned factors, we consider an archetype-based arrangement as a potentially successful approach to computationally represent and manipulate sound timbres in a human-like fashion. Furthermore, we consider fundamental for the characterization of a sound archetype the individuation and modeling of the features that are common with different instances of the archetype. Nevertheless, this assumptions has not to be intended as a generalizable strategy to algorithmically replicate human auditory-related intellection and imagination in all its aspects. Indeed, the particular approach proposed in this work is focused on the implementation of a sound synthesis and processing technique, precisely aimed at manipulating sounds by administrating perceptual audio features.

2.2 On features extraction

Feature is a misleading term since it could refer to concepts with slightly different nuances. On the one hand, it can be adopted as simple synonym of "characteristic". On the other hand, it has a precise scientific meaning, indicating particular qualities that can be algorithmically analyzed and extracted from data, and hence, from audio information. From here onwards we will primary use this term referring to the second connotation. Feature extraction techniques are widely employed in Music Information Retrieval (MIR)[3]. They permit to reduce the dimensionality of raw audio data, extracting only certain information that is meaningful for a precise task. This procedure is aimed at restricting the required computing resources and simplifying the algorithmic implementation of MIR applications.

In this context, it is relevant to classify features according to a hierarchy relative to their *abstraction degree.* Low-level ones are intended as simple signal-level properties, such as frequency and amplitude, whereas high-level ones are more sophisticated structures, for example music genre or even *beauty* or *sadness*. This hierarchy reflects also the *measurability level* of features. Indeed, low-level ones are usually precisely computable and the more abstract ones often can not be absolutely estimated. An important property of this organization is that high-level features can be described as *function* of lower-level ones, defining *sub-feature dependencies*. For instance, algorithms aimed at music genre classification are often based on beat tracking[4] [6]. The latter is in turn function of onset information[5], which is dependent on amplitude macro-variations of the signal's amplitude. This

3 Interdisciplinary science of retrieving information from audio signals. Some of the disciplines involved in MIR are: signal processing, musicology and psychology.

4 This feature describes the tempo and the beat morphology of an audio signal.

5 This feature describes the beginning position of relevant events in an audio signal.

(simplified) example reflects the increasing amount of complexity and arbitrariness that occurs by rising the features scale, starting from a precisely-estimable signal-level measure (amplitude) and ending with a complex quality such as song genre (that can be arbitrary even for humans). From a different point of view, audio features can be distinguished in static and variable or, in other words, time-invariant and time-variant. Even though audio signals are intrinsically time-dependent entities, an audio feature is considered static if it is not direct function of time, thus it is significant to measure it in any portion of a signal. Conversely, a variable one depends explicitly on time and then consists of the *mutation* of sub-features or acoustical properties. This distinction is an important aspect to consider in this context because feature extraction algorithms and Artificial Neural Networks can be highly sensible to the time-variance aspect, as will be discussed further on.

Feature extraction is a fundamental procedure for the simulation of human mind's capability of classification and re-evocation of perceptual experience. In fact, gestalt laws of grouping point out that human mind has an inner predisposition to extract patterns (features) from perceptual stimuli and rely on these patterns (instead of the raw representation of the stimuli) to categorize and recall experiences [7]. This phenomenon is based on the recognition of an isomorphic similarity between the perceived stimulus and psychological archetypical ideas (such as geometric concepts) that our intellect constructed through the experience. Thus, human mind performs an intellection that can be compared for many aspects to algorithmic high-level feature extraction [8].

One possible approach to perform feature extraction is to "manually" identify mathematical structures that represent the target qualities. For this reason we can call this practice *Handcrafted Feature Detection* (HFD). This approach has been extensively used to date. One notable example is Essentia [9], which is a state of art set of HFD tools developed at Pompeu

Fabra, Barcelona. Essentia demonstrates that this method can achieve a consistent precision in signal classification tasks. Nevertheless, a substantial drawback is its high specificity. Indeed, in most cases, this approach requires to separately model every single feature, building specific algorithms. HFD is based on the classic computing paradigm that involves to solve a problem following specific instructions contemplated and codified by the programmer. This limits the problem-solving capabilities of a software to problems that humans already know how to solve. Accordingly, this approach is not particularly suited for high-level features extraction tasks, since it can be difficult for humans to identify accurate mathematical structures to represent perceptual and arbitrary features [10]. Literature shows plenty of studies aimed at HFD high-level feature extraction. On notable collection of researches regarding strategies for physical-modeling of perceptual features is *Sounding Objects (SoB)* [11]. The methods presented in this work point out that HFD is a valid approach for high-level feature modeling, nevertheless confirm its high specificity. In fact, the majority of papers collected in SoB demonstrate remarkably accurate results, although in relatively restricted contexts.

A contrasting approach to perform high-level and human-oriented feature extraction is to adopt a generalizable data processing paradigm to replicate how human intellect interprets perceptual stimuli. This permits to extend the problem-solving capabilities of HFD methods, conferring an algorithm the ability of autonomously finding solutions to problems, including issues that humans do not know exactly how to resolve [10]. *Deep learning* strategies are oriented towards this direction, relying on a statistical model that replicates the information processing modalities of the biological nervous system: the *Artificial Neural Network* (ANN). For en exhaustive explanation of deep learning and ANNs refer to *Goodfellow et al*. [12] and *D. Kriesel* [13]. ANN's problem-solving abilities are based on

the *experience*, such as occurs for humans[6]. Therefore, they need to be *trained* with example data in order to be able to solve a problem. ANNs are capable of analyzing any kind of information (for example sounds, images and videos) in a similar manner as human brain does and perform complex operations among data, such as finding similarity patterns [10]. This confers an ANN the capability of performing human-like sophisticated operations such as data cataloguing basing on abstract criterions, for instance determining if an image portrays "happy" or "sad" people. A notable example of these capabilities is YouTube's video recommendation system [14]. ANNs can be viewed as algorithmic structures that follow rules analogous to the gestalt laws of grouping[7] [8]. This property makes an ANN highly efficient in recognizing perceptual and abstract features, which are often too complex to be mathematically expressed "by hand". Accordingly, the ANN can be considered as a commensurate approach to replicate the procedure adopted by the human brain for associating categorical attributes and metaphoric sensations to sounds, as suggests, among many others, the work of *Gounaropoulos et al.* [15].

ANN's training process affines their performance for a specific task. Thus, relying on the given experience, it makes the ANN an *expert system* for that task, providing the *knowledge* required to solve a precise problem. Therefore, ANNs are able to improve their accuracy as they processes data: the more data is analyzed, the higher quality of the results is, reflecting human learning modalities [10]. On our specific case, the required task is to build a model of perception-related sound features: sound archetypes. The given experience for this purpose has to be a set of sounds that are labelled

6 For example, humans learn the correct movements required to ride a bicycle by trial and error. In this case the experience consists of all movements performed during the training, associated with a memory of their effectiveness.

7 This aspect will be further discussed in the "Introduction to Convolutional Neural Networks" chapter.

by humans with the perceived level of the features[8]. Then, the system would find similarities among the files to produce a model that reflects "how a sound should be shaped" to match one particular perceptual characteristic.

For high-level feature extraction tasks, the ANN approach manifests two significant advantages compared to HFD. Firstly, ANN-based models are non-specific, providing the possibility to extract many different features with the same architecture, according to the given experience. Moreover, they are notoriously convenient for fuzzy and perceptual-oriented characteristics. On the other hand, this approach presents two fundamental drawbacks. First, ANNs must be trained with a large amount of data in order to reach a reasonable experience, and then precision. Moreover, consistent computing resources are often required because of the considerable amount of data to process. By examining these aspects, we focused only on the Artificial Neural Network approach to perform perception-oriented feature extraction. Even though there are many other plausible and potentially successful methods to achieve our task, we estimated this approach as the most suitable for our particular project. Notwithstanding the foregoing, we consider the possibility of performing comparisons between different strategies to be observed at a later stage.

8 This makes possible a supervised learning strategy.

3 TECHNICAL BACKGROUND

Two sub-categories of Music Information Retrieval are of strong interest for this project: *automatic signal classification*, based on the semantic content of sound files, and *feature matching data synthesis*, involving the generation of sounds that present specific target features. Furthermore, this research takes into account several studies concerning sound synthesis aimed to match specific *perceptual-oriented* features, which often involve both the above-mentioned fields.

3.1 Automatic signal classification

This practice concerns the algorithmic cataloguing of audio files according to their content. This is intrinsically related to features extraction, as a matter of fact it could be said that the two procedures coincide in many aspects.

It is possible to catalog audio signals according to different abstraction levels. In certain circumstances it could be valuable to adopt simple signal-level characteristics as descriptors. For instance, a database containing recorded violin tones could require to be organized according to the base pitch of the samples. Being the pitch a precisely computable feature, this task could be conveniently accomplished through HFD algorithms such as t h e *autocorrelation* [16]. Conversely, in other contexts it could be indispensable to adopt more abstract criterions for the classification. For example, an interactive song database (e.g. Spotify[9] and Last.fm[10]) can have

9 www.spotify.com

10 www.last.fm

sophisticated exigencies, such as the automatic selection of songs that are similar to the most liked by a user. Currently the leading technique for this specific purpose is *Collaborative Filtering*[11], which is adopted, among others, by Spotify and Netflix to recommend new media. This technique is based on usage data and tends to not suggest unpopular material, introducing significant biases in the predictions [17]. Conversely, ANN-based methods have been identified as more accurate and generalizable for this purpose [18]. Besides this specific case, various studies proved that, for automatic signal classification tasks based on abstract features, ANNs can outperform traditional algorithms based on *handcrafted feature extraction*[12]. The difference is particularly evident for applications that require large amount of data to be analyzed [19]. *Choi et al.* [20], for instance, implemented a remarkably accurate design to detect song similarities basing on Recurrent Neural Networks.

With an ANN-based approach, similarities among data-points are usually identified comparing a set of sub-features extracted from the raw data. The sub-features can be manually described through HFD techniques or can be identified by automated processes such the *LFE algorithm* implemented by *Nargesian et al.* [21]. ANNs provide the possibility of automatically extracting complex patterns from a sub-features-set[13], identifying *superstructures* (high-level features) that can be problematic to be determined by humans. Nevertheless, a drawback of this procedure, is that the ANN-learned features are difficult (in most cases impossible) to be clearly interpreted by humans. Indeed, despite the average precision of the

11 https://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify/22-Alternating_Least_Squarescode_httpsgithubcomMrChrisJohnsonimplicitMFMonday_January

12 It is important to mention that, before the popularization of ANN-based signal classification techniques, the reversed Hidden Markov Model was a common methodology in this field. Although we do not take it in consideration for automatic classification purposes, since it has been extensively proved that it is an obsolete approach, compared to ANNs.

13 Or even directly from raw-data.

results obtained with ANNs, *Pons et al.* [22] have identified a deficiency in this approach for sound classification tasks, which is caused by the lack of an accurate "*musical coherence*", to use their own words. In fact, such methods often perform as *"black boxes"*[14], which can not guarantee a precise control of what occurs on the inside. This is due to the overly generic character of ANNs and to the lack of clearly interpretable mathematical representations of the learned features. This research points out the importance of adopting a *motivated* architecture, fine tuned to produce results that can be clearly understood by humans. In fact, a combination of handcrafted and ANN-based feature extraction is proved to be an effective method to perform high-level feature extraction tasks. This procedure permits to focus the ANN learning on *motivated* and *task-related* characteristics of data, producing more interpretable (and accurate) outcomes. The work of *M. Stamenovic* [23] is an exemplary case of this trend.

Several ANN architectures are possible to perform data classification and, usually, different categories of tasks require different designs. Various studies proved that *Convolutional Neural Networks* (CNN) and *Recurrent Neural Networks* (RNN) are the most appropriate for audio applications. For a detailed overview of these two models refer respectively to *D. Stutz* [24] and *H. Jaeger* [25]. In particular, it has been proved that CNNs perform better for tracking static features, hence they are particularly suited for image-related applications[15]. On the contrary, RNNs are more convenient for time-related dependencies, therefore they are more convenient to model sequential data [26]. Thus, the choice of the ANN architecture is largely influenced by the temporal dependencies of the data and the features to be extracted. RNNs could seem the obvious choice for

14 A black box is a system that can be observed only for its inputs and outputs, without any knowledge of its internal functioning.

15 Accordingly, CNNs are suitable to deal with spectral representations of audio information.

audio-related applications, considering the intrinsic sequential nature of audio information. Although, *Zhang et al.* [27] demonstrated that an entirely CNN-based architecture can perform with comparable accuracy for audio classification tasks (speech recognition), providing a significantly higher computing efficiency than RNNs. Moreover, a combination of the two architectures has been ascertained to merge the benefits of both strategies, notwithstanding the high computing requirements. *Choi et al*. [26], for instance, have successfully adopted this approach.

3.2 Feature matching data synthesis

This procedure involves the generation of plausible data that present specific required features. In the field of audio processing, the majority of studies focused on text to speech synthesis applications. In the context of this research, it is targeted to the generation of sounds that are associable to specific sound archetypes. The literature shows two different approaches to achieve this tasks. The first consists of defining through learning algorithms the features to be emulated and generating the output audio data through computational architectures specifically arranged for sound synthesis, such as the *Sinusoidal Plus Noise Model* [28]. Instead, the second contemplates the direct synthesis of the final waveform through the learning algorithms. The learning-capable models that have been mostly adopted for feature matching data synthesis are *Markov chains* and *Artificial Neural Networks*.

The Markov chain is a stochastic process aimed at generating semi-aleatory sequences, constructed by procedurally recomposing existing examples belonging to the same complexity. Such a system can be trained to produce progressions emulating the behavior of given sequences, but adding random *coherent variations*. In other words, a Markov process is

capable of generating sequential data imitating the "style" of other data. This ability makes the Markov chain a convenient method to produce, for example, musical compositions (sequences of notes) that mimic other compositions or present an "organized aleatory" structure. In addition to this, Markov processes have been extensively adopted to perform timbre-level audio synthesis, generating the parameters needed by specific sound synthesis architectures. In fact, the set of parameters required by any algorithm could be intended as a (eventually non-temporal) sequence. *Hidden Markov Model Vocoders (HMMV)* represent the most common utilization of this technique. This procedure is based on synthesis techniques that simulate the vocal emission, defining *excitation* and *spectral* parameters. A HMM learns which parameters connote different phonemes, analyzing example values contained in a speech database[16]. After this training process, the model is capable of generating plausible parameters-sets that make the synthesizer produce the emulation of desired speech words [29]. The same concept can be adopted to control different synthesis models, such as additive, FM or granular architectures. Illustrious applications of stochastic Markov processes for granular synthesis (and music composition) are described in *Iannis Xenakis's Formalized Music* [30].

Markov processes and Neural Networks[17] have been used to perform strictly similar tasks to date[18]. However, besides other substantial contrasts, Markov chains are capable of generating only *sub-sequences* of the training data, that means concatenating portions of sequences that are present also in their experience. On the contrary, ANNs are capable of synthesizing pieces of data that are completely different from the training examples [31]. Thus,

16 Actually, from specific features extracted from the speech sound files.

17 In particular Recurrent Neural Networks.

18 In fact Hidden Markov Models have been extensively adopted also for automatic speech classification tasks.

ANNs are able to produce more dynamic and various outcomes than Markov chains, in feature matching data synthesis contexts.

Also ANN-based feature matching synthesis approach can be adopted for the definition of the features to be transformed in audio, whereas other specific algorithms synthesize the final waveform. *Zen et al.* [32] demonstrated that this technique can slightly surpass the accuracy of previous state of art methods, which were based on Markov processes. On the other hand, ANNs are proved to be enough powerful to directly compute the output waveform sample by sample. To our knowledge, Markov chains have never been adopted for similar applications. Google, with *Wavenet* [33], has proposed an important example of this technique. A crucial development of this implementation is that it has been trained to generate both speech and pianoforte audio files. It has been assessed through human judgements that this approach improves by 50% the previous state of art of speech synthesis in terms of naturalness [33]. Instead, generated piano samples have not been evaluated through formal surveys, although they are audible on the website[19] and, to our opinion, present a sharply realistic character despite some unwanted noise. In addition to this, Google has recently (in 2017) released a novel Wavenet-like encoder, based on a large dataset of sampled musical notes, which is called Nsynth [34]. Through this technology, Google achieved reliable models of several tonal and percussive instruments, implementing a software capable of reproducing and morphing realistic instrument-like sounds. However, it is important to denote that the experience needed to obtain these results consisted of circa 300000 4-seconds samples and the training required high performance hardware (and relatively long computing). Accordingly, these aspects make the Wavenet-like technologies problematic to be exploited with restricted resources, as occurs in the context of this thesis research. Another relevant experiment of ANN-based data synthesis has been proposed by *Reed et al.*

19 https://deepmind.com/blog/wavenet-generative-model-raw-audio

[35], demonstrating that a similar approach could be successfully adopted also for text to image synthesis.

Notwithstanding the average better accuracy of ANNs for the more complex models, the Markov chain approach shows a comparable performance in contexts with limited training datasets. Furthermore, Markov models permit to spare considerable computing resources, compared to ANNs. [36].

3.3 Perceptual feature matching data synthesis

Several techniques targeted to synthesize sounds matching specific abstract and perceptual features have been developed. One of the most notable is the *Concatenative Granular Synthesis* (CGS) [37]. Through this method, a feature is supposed to be matched by concatenating and overlapping fragments of sounds that are known to contain that particular feature. A valuable intuition on which this technique is based is the classification of sound timbres inside a multidimensional space, on which each axis represents a specific feature. By this representation would follow the important opportunity to re-synthesize sounds matching the mix of features present in one point of the space, and then to generate sounds matching morphed classes by moving on the features plane. Nevertheless, despite the potentialities of this technique, the granulation process involved in CGS often produces artifacts that make impossible to faithfully match a desired feature or sound archetype. Moreover, the process of concatenation is not suitable to reproduce time-variant features, which connotation extends beyond the fragments' duration. Accordingly CGS is not appropriate to reproduce time-related perceptual features.

A study strictly associated to our research has been undertaken by *Gounaropouls et al.* [15]. They described a method to classify timbres according to perceptual labels (e.g. *metallic*, *wooden*, *bright)* and synthesize sounds matching desired characteristics. This has been accomplished adopting 2 different Neural Networks: one for the classification of audio files and one for the synthesis of the parameters required by an additive synthesis algorithm.

A different approach for a similar purpose has been proposed by *Rocchesso et al.* in the above-mentioned *Sounding Objects* [11] collection, in which are presented several physical-modeling techniques to synthesize sounds matching specific perceptual features. One notable example is *B.L. Giordano*'s algorithm to categorize sounds according to the source material and synthesizing impact sounds matching "material macro-categories".

3.4 Background summary

Overall, audio signal classification is a consolidated approach and, in particular, ANN-based methods reached a consistent accuracy to date. Furthermore, the latter is an eminently active research and it is quickly becoming the attention summit of many audio processing developments. For audio classification tasks CNNs and RNNs have been proved to provide the best accuracy. Additionally, CNNs are demonstrated to be more efficient with regard to the computing requirements. Conversely, feature-matching audio synthesis has not been investigated at the same level, besides, in the majority of cases, researches have been focused on speech synthesis applications. In this context, Markov chains show a precision comparable to RNNs, especially for models that require restricted datasets. Moreover, they permit a simpler implementation and provide a significant

economy in terms of computing power demand. Besides, RNNs can produce more various outcomes, being capable of generating sequences completely different from their training examples. ANN-based perceptual feature matching synthesis is still a relatively new domain, although the above-mentioned researches (and many others) suggest that archetypical sound synthesis is not only possible, but can lead to satisfactory results. However, in spite of the encouraging outcomes, this concept has been rarely applied for artistic purposes beyond experimental contexts. Moreover, to our knowledge, researchers have not deeply focused on modeling strictly subjective perceptual sound qualities, that can be intended as high-ambiguity archetypes. We consider this particular aspect an interesting and potentially fruitful investigation.

4 METHOD

From a practical point of view, the objective of this research is to produce a working and usable framework to perform perceptual feature matching audio analysis and re-synthesis, relying on learning-capable models trained with a restricted set of observations. The goal is to obtain an optimized environment capable of being employed in real-time on a common laptop computer.

In this place, operational constraints have been fixed a priori, in order to obtain concrete and usable results in a restricted scenario. First of all, we focused on the concept of *sound texture*, intended as time-homogeneous aggregate of similar acoustic events that can be analyzed with "time-averaged statistics" [38]. By its definition, a texture contains sound qualities that can be considered *constant* over time, being time-invariant features or presenting an averageable time-variance. Conversely, non-texture sound events can present strictly time-variant characters. For instance, the timbre of a gong hit clearly evolves over time and most of its features can not be studied with time-averaged statistics. Thus, the sound texture presents an inner simplicity compared to "simple" sound objects, which makes it an effective vehicle to investigate human sound cognition, as pointed out by *McDermott et al.* [38]. Then, the whole research refers to the timbre dimension of audio information and does not take into account any possible implication derived from musical organization of sounds, which could be intended as complex and non-time-averageable organization of multiple sound events.

Another important consideration is that perception-related phenomenons can be studied following two radically different approaches: the Helmoltzian and the Gestalt-oriented view [5]. The first focuses its attention on the identification, quantification and interpretation of

neurophysiological processes derived from perceptual stimuli. Instead, the second is oriented towards the interpretation of sensations produced at a *psychological level* by the same stimuli, which are intrinsically non-absolutely-measurable entities and can be analyzed only through human descriptions. We consider the latter as a more efficient way to achieve our task, since sound imagination concerns abstract emotions that could be arduous to be mapped at a neurophysiological level (and we do not own the necessary competences to perform such investigation). As a corollary of adopting a non-exact methodology we are conscious to deal with a series of ambiguity factors that are intrinsic to human perception (and interpretation of perception). The most influent ones are:

- The interpretation of sensorial stimuli is influenced by the experience, culture and ethnic background of an individual;
- The interpretation of a stimulus can differ for distinct persons and can be ambiguous even for one single individual;
- There are many causes that can distort perception, such as the non-optimal operation of sensory organs;
- An individual could not be able to properly describe a perceptual sensation.

Furthermore, according to several lines of thought, part of the archetypical ideas that that are involved in human auditory intellection and imagination could not be derived directly from an individual's experience. For instance, they could be intended as innate concepts of our psyche, as part of a "collective unconscious". An illustrious study about this principle can be found in many works of *Carl Gustav Jung*, in particular *Children's Dreams: Notes from the Seminar Given in 1936-1940* [39] Nevertheless, this distinction is beyond the scopes of this research, thus, for simplicity, it is not examined in this context.

Finally, in this particular instance we consider to analyze and model only one sound archetype: the human perception of *chaos* and *order* in

audio information. We selected this characteristic in particular for its intrinsic fuzziness and non-specific connotation (compared to other features, such as *metallic*/*wooden*). Therefore, in order to achieve a faster and deeper comprehension on what will follow, we encourage the reader to take a moment to think about what determines a sound to be chaotic and which characteristics should have to be ordered, basing on his personal perception and experience. Obviously, there are not correct or wrong answers, being a strictly subjective conception. In fact, one the objectives of this project is to assess if different individuals conceive this feature in a similar manner, in order to define the reliability and generalizability level of its algorithmic model. The chose of modeling only one sound archetype is due to mere reasons of time. Indeed, our work is focused in demonstrating the practical potentialities of our method in the perspective of future developments. This makes our particular approach non-generalizable, since we adopted stratagems that are not proved to be appropriate for different perceptual sound features, especially the implemented dataset augmentation techniques, which will be explained in the next paragraphs.

The workflow we followed to obtain our archetypical analysis/re-synthesis framework is divided in four consecutive stages:

1. Dataset creation;
2. Development of the analysis algorithm;
3. Development of the re-synthesis algorithm;
4. Development of the user interface.

This path has been fixed a priori, considering to adopt an ANN based approach for the analysis and a Markov chain based architecture for the re-synthesis. This choice is supported by the argumentations exposed in the background section of this document.

4.1 Dataset creation

The training dataset can be regarded as the experience of an ANN. Analyzing and comparing the data-points, the ANN extracts *superstructures* (features) that reflect certain characteristics that are common with different data-points belonging to the same category. Through this procedure, an ANN achieves the ability of predicting if new data[20] matches a particular category, by comparing its superstructures with the ones learned through the training.

To obtain this behavior, we opted for a *supervised* learning method. This technique requires every data-point of the training dataset to be associated with a label that describes its target class (which, in this this context, coincides with a numerical value that indicates the perceived order level of a sound). This provides a clear landmark around which an ANN builds a model for a specific data-category. The homogeneity of the training dataset is a crucial aspect to obtain reliable predictions. In fact, it has been proved that non-homogeneous datasets (containing largely different amounts of data-points for each classification label) tend to produce unbalanced and inaccurate outcomes [40]. For this reason we have paid meticulous attention to collect a dataset as more balanced as possible.

As first stage, 100 sound have been downloaded from the Freesound online database[21]. The files have been randomly chosen through a simple aleatory algorithm built upon the site's API. This stratagem served to minimize the possible bias relative to our personal influence in the selection. Successively, the samples have been processed through a granular synthesis algorithm, in order to extract a large amount of different textures from each single sound. Without going into meticulous details, granular synthesis permits to segment an audio file into short slices (in the order of

20 Unobserved during the training.

21 www.freesound.org

few milliseconds) and re-combine them to obtain different timbres. Several elaborations can be individually applied to the sound slices, such as pitch/speed alteration, windowing and overlapping. Furthermore, multiple slices can be layered at the same time, producing *clusters* of short sound events. Through this procedure can be efficiently obtained texture-type timbres, intended as *McDermott et al.* [38]. This technique has been extensively experimented in electronic music contexts to date. For a detailed overview, please refer to *Curtis Roads' Microsound* [41]. We implemented this synthesis model through the software Max Msp. The core algorithm is represented in Figure 1.

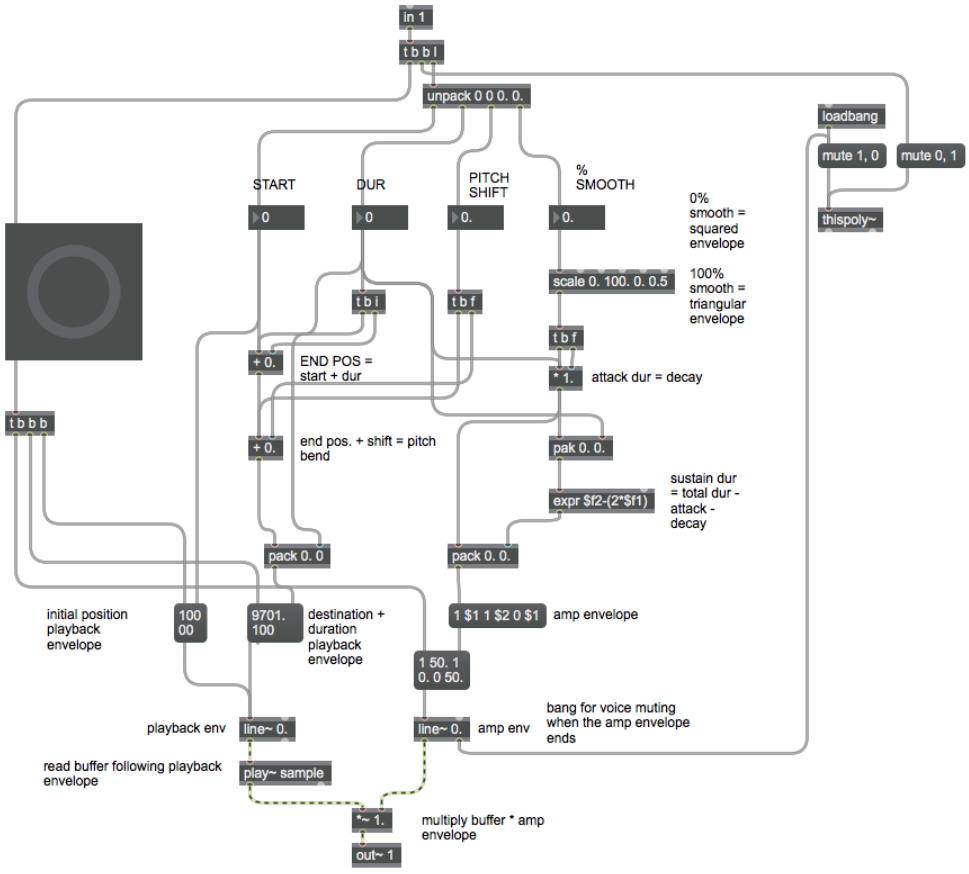

Figure 1: Core algorithm of the granular synthesizer

This patch extracts and reproduces one single slice from a sound file, providing control over:

- Initial reading point (referred to the source sample);

- Slice duration;
- Playback speed (which introduces pitch shifting);
- Envelope type (a continuous control: from rectangular to triangular).

600 instances of this algorithm run at the same time, triggered by a metronome, producing a cluster of overlapped sound particles, extracted from an audio sample. The parameters of each instance are determined through an aleatory algorithm that permits to randomly flutter its setting around a fixed value. The output results are texture-type sounds, which timbre content and morphology depend on the source sample and the synthesis parameters. The parameters-set that identifies a texture consists of an array of 9 values:

1. Source sample ID (referred to the sounds downloaded from Freesound);
2. Metronome speed;
3. Initial slice position;
4. Position randomness;
5. Initial slice duration;
6. Slice duration randomness;
7. Initial playback speed;
8. Playback speed randomness;
9. Envelope type.

A collection of 1000 3-seconds sounds has been recorded[22], trying to obtain a balanced dataset**.** In order to reduce possible influence in the dataset collection process, the parameters-set creation has been committed to another aleatory algorithm, which produces random parameters-sets on demand. We empirically denoted[23] that the algorithm tends to generate sounds unbalanced towards the perceptual chaos. For this reason, a hierarchy of 5 discrete levels (classes) of the inquired feature has been

22 With a sample rate of 44100 Hz and a bit depth equal to 16.

23 This has been assessed through informal judgements of a dozen of subjects.

defined and one single person manually created randomly tuned textures and selected 200 sounds for each class. This avoided to obtain a collection containing a highly larger amount of chaotic sounds. Therefore, it has been necessary to impose decisions based on the perception of a single individual. This certainly introduced bias, although we retained this aspect secondary to possibility of obtaining a dataset composed of almost only chaotic sounds. The person who created the dataset was 28 years old, has never had auditory dysfunctions and studies and practices electronic music and sound design.

4.1.1 Dataset human classification

An accurate labeling of the data-points is a necessary procedure to permit a supervised learning architecture. To obtain human judgements concerning the perceived chaos/order level of the recorded textures, an individual survey has been proposed to 80 distinct subjects. The test has been planned as an interactive electronic document that outputs a text file containing all the responses. The algorithm has been implemented through the software Max Msp. The primary target was to obtain 4 different classifications for each sample, to be able to perform significative statistics among judgements.

The test consisted of 3 consecutive sections:

1. General and attitudinal questions;
2. Sounds' classification;
3. Adjectives matching.

Every section is provided with a clear explanation of the tasks to be accomplished, as well as instructions for the interface's usage. Every test

proposed the same questions to all subjects and 50 different[24] sounds to be evaluated. The survey has been written in the italian language. No time limits have been imposed, although the duration of the tests oscillated approximately between 15 and 20 minutes. We tried to recreate strictly similar conditions for each instance, adopting the same laptop (Macbook Pro 2011), the same headphones (Beyerdynamic DT 770 PRO) and proposing it in relatively quiet rooms. Furthermore, the first 10 instances served also to verify the correct functioning of the algorithm, although, since all worked properly, no modifications have been applied.

To select which sounds to propose in each test, a numeric ID has been associated to every texture contained in the dataset (from 0 to 999) and to every test to be performed (from 0 to 99). In order to minimize any possible bias, the list of samples has been scrambled, generating an array containing 4000 random integers, that is the total number of sounds to be classified[25]. The randomization process has been arranged to insert each sample ID in the list exactly 4 times. Then, the 50 sounds assigned to a test are identified scrolling through the indexes of the random IDs list. In particular, every test contained the sounds starting from (*current_test_ID* x 50) and ending with (*current_test_ID* x 50 + 49). By means of this procedure, 80 classification tests provided 4 independent judgements for every data-point. Furthermore, being the random list of IDs fixed, it has been possible to exactly replicate a particular test (containing the same sounds, in the same order). This served for re-proposing a test instance, in case of accidental damaging of the output file, or else for hearing the exact samples that a tester classified. A detailed description of the test's architecture follows.

The first section of the electronic document contains the succeeding questions:

24 Different one another within a single test and differently selected and sequenced among different tests.

25 4 independent classifications for 1000 sounds.

- Name and surname initials;
- Age;
- Have you ever had hearing dysfunctions?;
- Have you ever studied in depth sound or music related subjects?;
- Do you regularly practice sound or music related activities?

The first question serves to identify the person who took a particular test, to eventually re-propose it in case of damaging of the output files. Whereas, the others are targeted to diagnose possible bias-factors in the classification that could derive from the age or the personal background of the testers. The successive section initially proposes 3 short previews of the sounds to classify. A preview consists of a selection of 5 concatenated timbres taken from the classification dataset, individually separated by one second of silence. Every preview contains sounds from different order classes[26], randomly selected and sequenced to avoid to impose any influence based on our prior classification. This procedure serves to give a preventive idea of the timbres that the tester will have to classify, reducing the possibility of biasing the judgement of the first samples. During this operation is also possible to adjust the listening volume, which then remains fixed until the end of the test. After this stage begins the actual sound classification stage. The user interface of this section is portrayed in Figure 2. This interface permits the user to jump forward and backward in the space of the samples to classify, providing the possibility of changing previously given responses and momentary skipping sounds. Moreover, a sample can be played as many times as the tester needs. This stratagem is aimed at avoiding casual or hasty responses, due to possible lacks of attention. The classification is organized as a discreet series of 11 mark boxes, of which only one can be checked, identifying an ascendent Likert-type scale [42]. We selected to adopt a discreet measurement to match the

26 The same classes established in the dataset collection process.

categorical design chosen for the automatic signal classification algorithm, as will be discussed later.

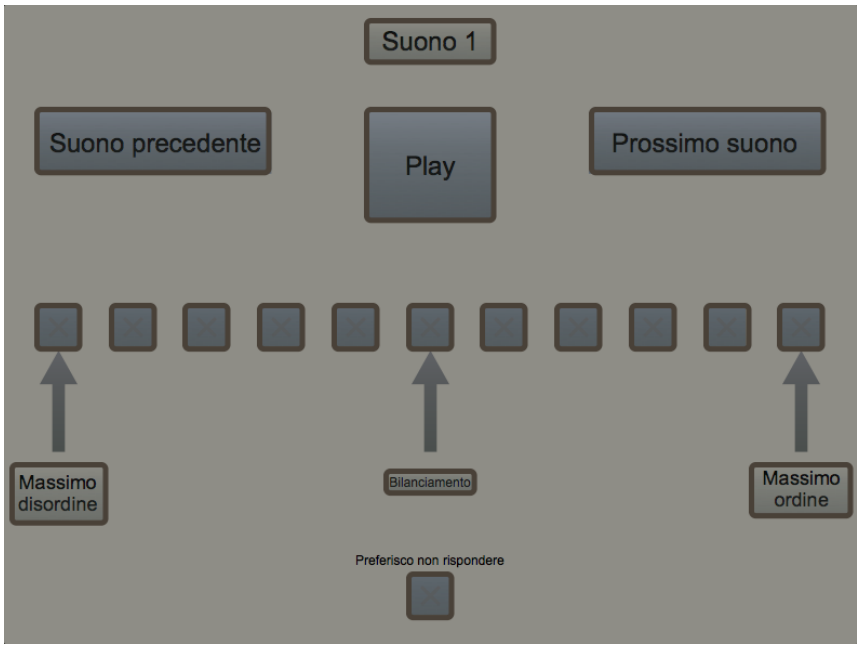

Figure 2: Classification user interface

In order to minimize possible biases that could derive from the visual layout, we adopted the following precautions:

- The boxes are equal, equidistant and centered in the test frame;
- The boxes are not associated with numerical values. Instead, qualitative references are given at the extremes and center of the scale (*maximum chaos, balance, maximum order*);
- For every test, the scale direction is randomly selected: from chaos (left) to order (right) or vice-versa (switching also the position of the reference labels).

In addition to this, since the inquired feature could be overly ambiguous for certain samples, the test provides the possibility to abstain from the judgement of single data-points. After the completion of the classification process, the tester is invited to review all the given marks from the beginning. This procedure is aimed at further reducing possible bias since,

during the test, the user's confidence with the task could increase, causing a possible shift of his judgement parameters. The final passage is optional and requests a tester to separately list attributes that he would associate to a chaotic and to an ordered timbre, basing on his personal criterions. This serves to identify perceptual sound archetypes associated to the chaos/order. Furthermore it gives a descriptive idea of the judgement parameters adopted by the testers.

4.1.2 Survey statistics

Due to the relatively long duration of the tests and the amount of needed instances, the dataset human classification process has been accomplished approximately within 2 months. Once collected the target amount of judgements, the results have been statistically analyzed, in order to define:

- The ambiguity level of the human perception about the inquired feature;
- Possible bias factors in the classifications, correlated to the age or the background of an individual;
- A homogeneous training dataset, containing an equal amount of sounds for every class, in which each data-point is associated with a univocal label that describes its averagely perceived order level.

In total, 3983 (over 4000) classifications have been performed, whereas 17 abstentions of judgement have been collected. All 80 testers are italian and resident in Veneto. The age range is spread within 19 and 62, with a mean age equal to 30.7. 76 testers declare to have a perfect auditory system, whereas 4 of them present auditory dysfunctions, which, in the majority of cases, consist of tinnitus. An amount of 28 testers assert to not have a musical background, neither regarding music or sound related studies, nor

concerning practical activities. On the contrary, 52 persons present a sound/musical background, among which, 34 regarding studies and practice, 13 concerning the only practice and 5 referred just to studies.

Even though the testers classified the data-points checking non-numbered boxes, for the result's analysis we associated the judgements to discreet numerical values (from 0 to 10). In this particular context, we retained appropriate to average the perceived order level of the single data-points according to the rounded arithmetic mean of the obtained classifications. Being the inquired feature a perceptual and culture/experience dependent characteristic, we considered opportune to weight also the outliers in the statistics. In fact, in this case, they could represent a valid, although detached, point of view. Therefore, the arithmetic mean, which equally weights all data, seemed to us a simple but effective choice in contrast to other average descriptors such as, for example, the median or the mode. Nevertheless, this increases the possibility of weighting results that could have been deviated by factors different from the mere perception of an individual, which include task's misunderstanding, hasty or random answers and errors relative to the user interface's usage.

This said, let us now turn to analyze the collected results, considering that we will refer to the order class of a data-point as its classifications' rounded mean. Furthermore, from here onwards we will refer only to a *chaos-to-order* scale, going from 0 (max chaos) to 10 (max order). Since the scale orientation has been randomly selected for every performed test[27], all the classification given for the inverse range have been opportunely rescaled to match this univocal representation.

27 Chaos-to-order or order-to-chaos.

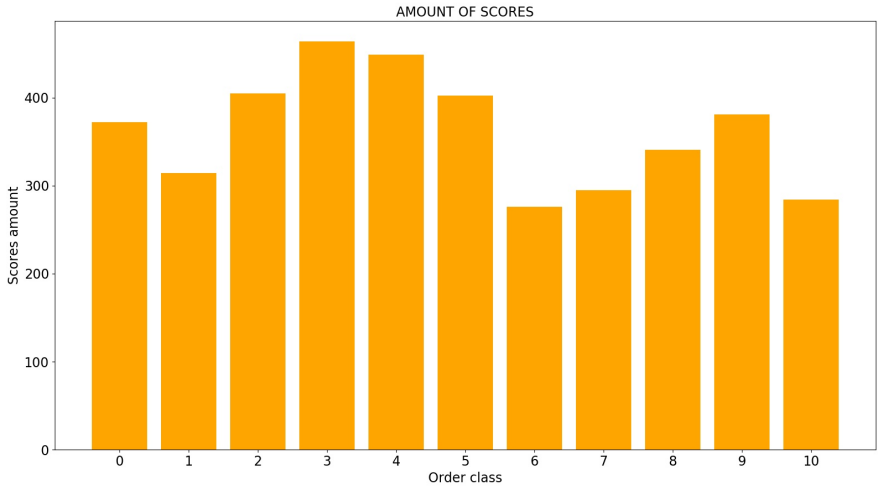

Figure 3: Amount of scores for each order class

Figure 3 shows the amount of collected classifications, distinguished according to the selected order class. From the chart is evident a disproportion among classes, going from a minimum of 276 scores given for class 6, to a maximum of 464 ones for class 4. This means that the initial dataset was unbalanced, presenting a disparity in favor of mid-chaotic sounds (from class 2 to 5) and a relative scarcity of samples belonging to class 6 and 10. From this inequality can be inferred that the inquired feature presents a certain amount of ambiguity and therefore it is differently perceived by distinct individuals, as we expected. In fact, if were not so, the amount of scores obtained for every class would have been more coherent, reflecting the prior classification performed by the person who created the dataset. A detailed portrait of the classifications' distribution is represented in Figure 4.

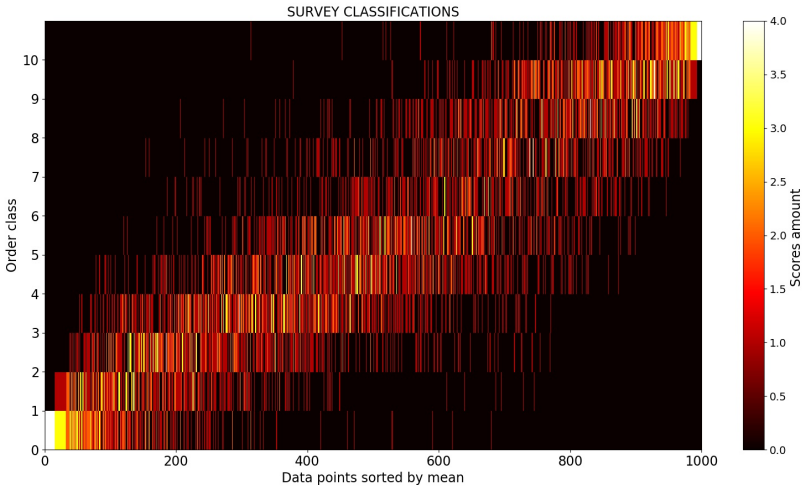

Figure 4: Collected classifications for each order class

In this chart, along the X axis are represented all the data-points, sorted by their classifications' mean (from 0 to 10, and then from chaos to order), in the Y axis are plotted all the given scores, arranged from 0 to 10 and the Z axis, which is represented by the color's brightness, indicates the amount of registered scores in a certain point. Therefore, each horizontal row shows the amount and the spread of the given classifications for a class. The graphic reveals (especially observing it from afar) that the scores given for the classes to the extremes of the vertical scale (0,1 and 9,10) are more concentrated, despite sporadic outliers. This indicates that people are more concordant in conceiving the concepts of extremely ordered and extremely chaotic sound. On the contrary, the perception of the intermediate classes of the same feature is clearly fuzzier, presenting sharply more dispersed classifications for the classes going from 2 to 8, with a peak of spread for class 7. In order to obtain a clearer representation of the ambiguity level of the perception of the feature, we computed the *mean standard deviation* for each class. This value is the quadratic mean of the standard deviations of the classifications of every data-point of each class. The standard deviation of each data-point is computed applying:

$$\sqrt{\frac{\sum_{n=0}^{N-1}(X_n - X_m)^2}{N}}$$

where X_n are the classifications collected for one data-point, X_m is the arithmetic mean of the classifications and N is the amount of classifications. Consequently, the mean standard deviation for a class is computed as:

$$\sqrt{\frac{\sum_{n=0}^{N-1}(X_n)^2}{N}}$$

where X_n are the standard deviations of the data-points belonging to an order class (with the same rounded classifications' mean) and N is the amount of data-points belonging to the class. This value represents the average dispersion of the classifications given for all data-points correlated to a class. Therefore it is directly associated to the concept of class-wise ambiguity level of the inquired feature. We selected this descriptor for its interpretation immediacy. In particular, because it is expressed in the same units as the distribution value, and then it can be directly compared with the classifications. Probably, other descriptors could be more accurate than the standard deviation for this purpose. Nevertheless, it is important to denote that the calculation of statistical dispersion within extremely little data (4 classifications for every sample) could never lead to absolutely precise and representative measurements. Accordingly, this measure can not be considered as an exact computation of the ambiguity level of the perceptual sound chaos/order feature, although we think it gives an enough-trustworthy estimation of it.

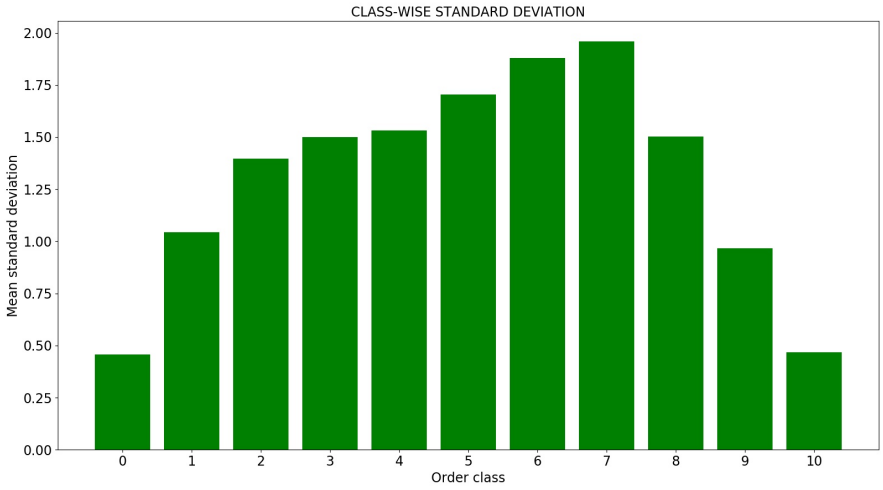

Figure 5: Mean standard deviation of each order class

Figure 5 represents the mean standard deviations computed for each order class. This graph clearly shows that the ambiguity level is sensibly lower to the extremes of the scale, confirming what we inferred from Figure 4. Moreover, it better reveals that the discordance follows a quasi-gaussian shape centered on class 7, indicating that the ambiguity gradually increases from the extremes to the mid-order area. This further validates the assumption that the concepts of extreme chaos and order are more equally perceived than the transitional degrees of the feature, with a maximum ambiguity excursion of almost 2 classes upwards and downwards for class 7[28]. In order to obtain an overall ambiguity level of the inquired feature, we computed the mean standard deviation among all data-points, obtaining a value of 1.483. Since the maximum standard deviation of a scale going from 0 to 10 is 5, by performing a simple proportion we achieved an indicative 29.6% of ambiguity. Nevertheless, also about this value, the above-stated considerations do apply.

28 This means that, within the collected data, the classifications of every data-point referred to class 7 (according to the rounded mean) commonly fluctuate between class 5 and class 9.

To identify possible bias factors, we identified several sub-datasets, isolating testers with selected age ranges, musical background or auditory dysfunctions. The mean standard deviation of each sub-dataset has been computed, comparing all the classification performed by a people's category with the classification's mean of each data-point relative to all classifications[29]. The obtained values reflect how much a specific people's category deviates from the overall trend. Figure 6 summarizes the obtained results.

AGE RANGE	AUDITORY DYSFUNCTIONS	SOUND/MUSIC STUDIES	SOUND/MUSIC PRACTICE	MEAN STANDARD DEVIATION	AMOUNT OF TESTERS
<= 26	any	any	any	1,567	31
> 26 <= 30	any	any	any	1,352	31
> 30 <= 40	any	any	any	1,606	7
> 40	any	any	any	1,521	11
any	yes	any	any	1,409	4
any	no	any	any	1,487	76
any	any	no	no	1,551	28
any	any	yes	yes	1,444	34
any	any	yes	no	1,39	5
any	any	no	yes	1,414	13

Figure 6: Sub-datasets' mean standard deviation

In general, the table shows that there are not significant differences in the perception of the inquired feature among the isolated sub-datasets. However, in particular for testers over 40, testers with auditory dysfunctions and testers with only academical musical background, the scarcity of observations could have deviated the results. The only reliable biases that emerge from this investigation are:

- Persons aged between 27 and 30 seem to be the closest ones to the average classifications, compared with the other age-related people's classes;

29 The mean is computed taking into account all people's categories.

- Individuals that present a strong musical background (both studies and practice) seem to be closer to the average than the ones who have neither studied, nor practiced music.

Nevertheless, it is important to consider that all the involved testers belong to a strictly similar culture background, being all italian and resident in Veneto. Accordingly, through this survey it is not possible to identify eventual culture-related biases in the human conception of the feature. Even though it has not been possible to implement an inter-cultural survey in this instance, we consider this investigation an important pursuance of our research.

Once achieved a clear, although limited[30] portrait of the human perception of chaos and order in sound information, we performed a direct comparison of this representation with the *entropy* computed for each classified sample. The entropy is a mathematical descriptor capable of expressing the amount of "unpredictability" and "chaos" contained in a vector. Accordingly, it could be considered as the "mathematical counterpart" of the perceptual feature investigated in this research. The concept of entropy was introduced by *Claude Shannon* in *A Mathematical Theory of Communication* [43], which is considered one of the most important foundations of the *theory of information.* We computed the entropy of the sound vectors adopting the Python function *scipy.stats.entropy*, which applies:

$$-\sum_{n=0}^{N-1} X_n(\ln X_n)$$

30 Limited to a restrict amount of people (80), belonging to the same cultural background and relative to 1000 texture-type sounds.

where X_n are the individual samples of an input signal and N is the total amount of the samples. This computation is capable of discriminating how much predictable (repetitive) patterns are present in the source. The obtained value expresses the *amount of information* contained in a signal, intended as the amount of events that are "distant" from the general trend of the source, and then unpredictable. The more a signal is predictable, the less information it contains. Accordingly, the higher is a signal's entropy, the higher are its intrinsic unpredictability, chaos and noisiness.

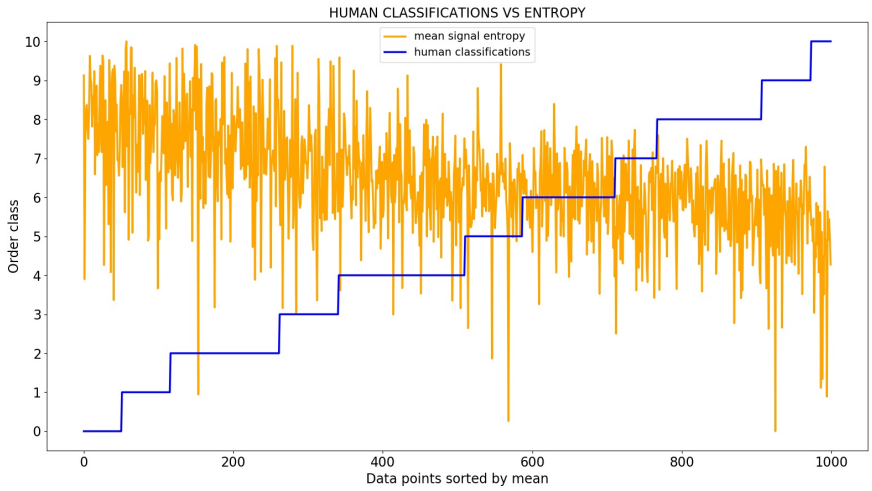

Figure 7: Human classifications vs. signals' entropy

Figure 7 displays along the X axis all the data-points sorted by their classifications' mean and on the Y axis the human classification's mean (in blue) and the entropy of the relative sounds (in orange), normalized in the same range of the classifications (from 0 to 10). From this representation it is evident that the entropy shape is sharply noisier, compared to the human classifications. Nevertheless, it is possible to recognize common tracts in the two. In fact, despite the heavy oscillations of the entropy, they present opposite general trends, respectively, averagely decreasing and increasing. Moreover, it can be recognized a certain coherence in the entropy shape

within areas correspondent to human perceived order classes (horizontal blue steps). Accordingly, standing on our survey and on the adopted data representations, it could be affirmed that entropy can not faithfully describe the average human perception of the chaos and order in sound information, although the two measures are certainly correlated.

Finally, in order to obtain a balanced dataset for the ANN training (with an equal amount of data-points per class), it has been unfortunately necessary to discard the 70.3% of the samples. The approved data-points have been selected according to the lowest mean standard deviation, eliminating the sounds with the more ambiguous classifications. The ultimate training dataset consist then in a collection of 297 sound textures (27 for each class), associated to their averagely perceived chaos/order level, obtained computing the rounded mean of the relative classifications.

4.1.3 Considerations on the perception of auditory chaos and order

The final section of the test (optionally) requested to list possible attributes correlated to chaotic and ordered sounds, according to the personal perception of the testers. Figure 8 and 9 display the collected results, sorted by the recursion amount of the adjectives. The original descriptions in the italian language are reported, as well as the corresponding english translations, obtained through a double check on the online dictionaries *Word Reference*[31] *and Reverso Context*[32].

31 https://www.wordreference.com

32 http://context.reverso.net

CHAOS ATTRIBUTES		
ITALIAN	ENGLISH	SUM
fastidioso	annoying	7
ruvido	coarse	6
caotico	chaotic	6
irregolare	irregular	6
rumoroso	noisy	4
discontinuo	discontinuous	4
confuso	confused	4
scostante	non-constant	3
incostante	non-constant	3
imprevedibile	unpredictable	3
casuale	random	2
scosceso	steep	2
distorto	distorted	2
duro	hard	2
inarmonico	non-harmonic	2
confusionale	confusional	1
difficile da capire	hard to understand	1
suono composto	composed sound	1
disagio	discomfort	1
scomposto	dismantled	1
spigoloso	Sharp-cornered	1
tanti suoni sparpagliati	a lot of scattered Sounds	1
destabilizzante	destabilizing	1
disconnesso	disconnected	1
articolato	articulate	1
vario	various	1
non ritmico	non-rhythmic	1
eterogeneo	heterogeneous	1
striato	streaked	1
spiacevole	unpleasant	1
disorganizzato	disorganized	1
crudo	raw	1
intermittente	intermittent	1
complesso	complex	1
dinamico	dynamic	1
privo di logica	without any logic	1
costante	constant	1
sminuzzato	crumbled	1
rigato	striped	1
rugoso	wrinkled	1
grigio	gray	1
discorde	discordant	1
confusionario	bumbling	1
scaleno	scalene	1
asimmetrico	asymmetric	1
variabile	variable	1
frammentato	fragmented	1
contorto	contorted	1

ORDER ATTRIBUTES		
ITALIAN	ENGLISH	SUM
regolare	regular	8
lineare	linear	7
costante	constant	6
piacevole	pleasant	5
liscio	smooth	5
semplice	simply	4
armonico	harmonic	4
pulito	clean	4
continuo	continuous	4
dritto	straight	3
limpido	clear	3
armonioso	harmonious	3
morbido	soft	3
fluido	fluid	2
stabile	stable	2
piatto	flat	2
compatto	compact	2
ordinato	ordered	1
singolo	single	1
soporifero	soporific	1
leggero	light	1
suono singolo e dritto	single and straight sound	1
non disturba	it doesn't disturb	1
discontinuo	discontinuous	1
ritmico	rhythmic	1
liquido	liquid	1
sembra un flusso continuo	it seems a continuous flux	1
placido	placid	1
equilibrato	balanced	1
coerentemente organizzato	coherently organized	1
coeso	cohesice	1
deciso	determined	1
simmetrico	symmetrical	1
ripetitivo	repetitive	1
statico	static	1
prevedibile	predictable	1
sidereo	sidereal	1
fatiscente	crumbling	1
lunare	lunar	1
rasserenante	calming	1
distensivo	soothing	1
setoso	silky	1
alto	high	1
rotondo	rounded	1
freddo	cold	1
cadenzato	lilting	1

zoppicante	limping	1
strascicato	slurred	1
singhiozzante	sobbling	1
angosciante	angsty	1
sporco	dirty	1
poroso	porous	1
tiepido	warm	1
ansioso	anxious	1
dissonante	dissonant	1
scoordinato	uncoordinated	1
invadente	intrusive	1
dissociato	dissociative	1
frastagliato	jugged	1
stonato	off-key	1
polifono	polyphonic	1
incoerente	incoherent	1
spezzettato	crumbled	1
molti timbri senza armoniche scoordinati nel tempo e nella durata	a lot of timbres without harmonics, uncoordinated over time and duration	1
stridente	strident	1
senza una logica ritmica	without any rhythmic logic	1
disturbante	disturbing	1
disorientante	disorienting	1
sbilanciante	unbalancing	1
non periodico	non-periodic	1
non uniforme	non-uniorm	1
incasinato	messy	1
intricato	intricate	1
sfocato	out of focus	1
impreciso	inaccurate	1

Figure 8: Attributes given for chaotic sounds

ripetitivo nel tempo	repetitive over time	1
misurabile	measurable	1
coerente	coherent	1
unito	united	1
melodico	melodic	1
comprensibile	comprehensible	1
suoni armonici ordinati nel tempo	harmonic sounds ordered over time	1
distinguibile facilmente	easily distinguishable	1
pensato	thoughtful	1
periodico	periodic	1
uniforme	uniform	1
disteso	relaxed	1
scorrevole	fluid	1
nitido	tidy	1
preciso	precise	1

Figure 9: Attributes given for ordered sounds

The most recursive attributes collected for chaotic sounds are: *annoying, coarse, chaotic* and *irregular*. Whereas the most frequent adjectives given for ordered timbres are: *regular, linear, constant* and *pleasant*. In general, a semantic coherence within the classes can be identified, despite 2 single attributes that come from the same test instance, which can be interpreted as a misunderstanding of the task: *constant* (for chaotic sounds) and *discontinuous* (for ordered sounds). As we expected, several attributes refer to the visual (*sharp-cornered, rounded...)* and to the

tactile (*coarse, silky...)* realms, indicating the presence of *synesthetic/cross-modal* implications in the perception of the inquired feature. Certain adjectives literally allude to a *negative* connotation for chaotic sounds (*annoying, discomfort...),* while several ordered timbres are associated to *positiveness* (*pleasant, calming...)*. The overall trend of chaos adjectives seem to point to the semantic sphere of *dysphoric,* while the ordered ones tend to the concept of *euphoric*. It is interesting to denote the recurrent reference to the idea of *continuity (linear, straight..)* and *discontinuity* (*crumbled, fragmented..)*. This suggests that several subjects interpreted the order level as *homogeneity* level of the interpenetration of the sound unities that constitute a texture. Moreover, a few observations consider the eventual *rhythm*ic character of the textures, valorizing the contrast between *repetitiveness* (order) and *variability* (chaos), applying an analysis correlated to the above-mentioned signal *entropy*. This distinction, united with the chaos/negative/dysphoric and order/positive/euphoric connotations, suggests a metaphoric relationship with the sphere of *sickness/wellness*. In fact, rhythmic and regular biological patterns (for instance heartbeat, breathing or circadian cycles) are associated to health. On the contrary, the disorganization of these patterns can lead to pathologies such as arythmia and insomnia, referring to the concept of disease. Furthermore, a minority of testers weighted a *strictly spectral* character of sound, identifying *harmonic* timbres as ordered and *inharmonic* ones as chaotic, even though this quality could be implicated also in other adjectives such as *annoying* and *pleasant.* This is connected with the concept of organization level, being the harmonicity a type of spectral structure that humans can recognize without any difficulty.

Accordingly, standing on what emerged from these subjective descriptions, the scale going from the perceptual chaos to the order seem to manifest as confluence of the textures' morphology towards an *organized structure* (as several given attributes suggest: *harmoni*c, *regular, continuous*

flux, coherently organized, lilting, predictable...). This interpretation is perfectly coherent with the information-theory definition of the pure chaos as absence of structure and of pure order as absence of information. In fact, the comparison between the human perception of sound order and the signal entropy of the classified sounds clearly revealed an inverse correlation between two, identifying chaotic sounds as averagely more unorganized structures than the ordered ones.

This triggered a suggesting interpretation, which should be taken just as a personal reflection. A chaotic system is interpretable as a deterministic system in which elapse non-linear and highly complex dependencies. These are difficult to be interpreted by humans and often are approximated to the concept of randomness. This happens in the every-day life, as well as in the scientific field. This approximation serves to humans to isolate and monitor what is not under their control and comprehension, conceptually shifting what is simply complex in something unpredictable. Therefore, the pure chaos does not exist, being actually extreme and incomprehensible complexity. The watershed that distinguish chaotic and ordered phenomenons can then be identified in the point in which humans surrender, the point beyond which we are no longer able to calculate and outline the structure of a phenomenon. Accordingly, following this point of view, chaos is *defeat* and order is *satisfaction* (for having understood) as, moreover, suggest the semantic area of many attributes collected with the survey. In fact, as it is empirically evident (and countless researches confirm), the perception of ordered patterns can give a sense of satisfaction (and vice-versa) also in visual and tactile contexts. These reasonings reveal an interesting suggestion about modeling the chaos/order sound archetype in particular: it permits to investigate how a *complex* and non linear system, as the human auditory perception, *interprets* the very concept of *complexity* and *non-interpretability*.

4.1.4 Dataset augmentation and segmentation

The prediction accuracy of an ANN algorithm is directly dependent on the dimension of the training dataset. In most cases, the larger is the given experience, the better are the ANN's outcomes, reflecting their learning-based behavior[33]. Usually, the datasets adopted for deep learning tasks can reach hundreds of thousands or even millions of data-points. For example, the famous MNIST dataset[34] counts 70000 images in total and the above-mentioned Nsynth comprehends over 300000 sampled sounds. Nevertheless, in certain situations it could be problematic, or even impossible, to collect such large data. In fact, besides other specific cases, when human labeling is mandatory, the time and resources required to collect large datasets could be consistent. On the other hand, undersized training data can lead to overfitting problems. This phenomenon occurs when a model adapts to the observed data, having an excessively higher complexity (number of parameters), compared to the amount of observations (data-points). This usually leads to an optimal accuracy for the data observed in the training process, associated with a significantly lower precision for new data. This makes a model ineffective, being unable to properly generalize the learned concepts.

Various strategies to reduce the overfitting have been developed. One of the most adopted is the *dataset augmentation* [44]. This method consists of generating series of "slightly altered" versions of every data-point, maintaining undamaged the features to be predicted. This permits to extend the size of a training dataset, and then to increase the accuracy of a deep learning model. For a visual classification task, for instance, typical augmentation techniques implicate stretching and rotation of the training

33 As occurs for human learning-based activities.

34 A dataset containing images of handwritten digits. It is usually adopted to test the accuracy of several deep learning architectures.

images. In our particular case, a cascade of spectral and time-related elaborations is applied to each sample, in order to create alternative versions of them, maintaining equal the original amount of perceived order level. This process has been entirely implemented through the Python language. The following processing algorithms are sequentially applied to one sound to produce one augmented file:

- Convolution between high-energy spectral areas. Initially, the STFT of the input signals is performed and the spectral peaks are detected. For this we adopted functions extracted from the SMS-Tools library[35]. After this stage, a random amount of the highest peaks (from 1 to 3) is convolved with other randomly chosen lower peaks. This process permits to obtain samples with different spectral shapes, maintaining the "spectral imprinting" of the original samples;
- Random filtering. A random amount (from 1 to 4) of 2nd order notch Chebytshev filters are applied to the input signal. Cutoff and Q are randomly tuned within a utile range. This generates randomly equalized versions of the original sounds;
- Random time stretching. A simple resampling-based time stretching algorithm is applied to the input signals, stretching (with pitch shift) the sounds by a random percentage (from 0% to 30%);
- Convolution with random impulse responses (IRs). Input signals are convoluted with randomly chosen room impulse responses, collected from the Voxengo website[36]. The balance between dry and wet signal is controlled by a random variable. This adds to the original signals the simulation of ambiance reverberation.

In particular, the random filtering and the IRs convolution have been fundamental to improve the model's accuracy for signals recorded with microphones, which can contain reverb and heavy equalizations due to the

35 https://github.com/MTG/sms-tools

36 http://www.voxengo/imodeler

microphone's and the room's characteristics. With this technique is possible to generate dozens of altered versions of every sample contained in the dataset. To ensure the diversity of each augmented data-point, the parameters and the sequence of the elaborations are randomly created for every instance. For this project we retained sufficient to produce 10 augmented files for each datapoint.

In addition to this augmentation technique, we implemented another stratagem to further enlarge the dataset's dimension. Every 3-seconds sound (including the augmented data) is segmented in 0.5-seconds frames, overlapped by 50%. The choice of simple rectangular windows has been empirically verified to lead to the most accurate results, compared with other common windowing functions such as triangular, hamming or blackman-harris. Since our pseudo-real-time implementation of the CNN-based audio analysis is based on a recording buffer as long as the training samples (as will be deepened further on), the segmentation provided a more fluent behavior. Considering that the data-points consist of texture-type sounds, and are therefore analyzable with time-averaged statistics, this segmentation process is supposed to not damage the original amount of the inquired feature.

Unfortunately, it has not been possible to organize a formal survey to assess whether the augmented (and segmented) files maintain the same perceptual order level of the original samples. However, this property has been verified through formal judgements of a dozen of individuals, with a positive feedback. Nevertheless, these augmentation and segmentation techniques can not be considered as a generalizable method. In fact, it is not proved that the resulting files maintain undamaged features different than the chaos/order.

At the end of the first research stage, we obtained a homogeneous dataset consisting of 297 sound samples, associated to the average human

perception of their chaos/order level. Furthermore, we developed 2 processing tools (augmentation and segmentation) capable of drastically increment the dataset's size. As will be discussed in the next chapter, we analyzed the effectiveness of these algorithms by training the classification model with and without the application of augmentation and/or segmentation to the dataset. The inquired feature has been proved to present an ambiguity level of 29.6%, relying on the tester's judgements and the adopted descriptor (mean standard deviation). Therefore, it can be considered a feature that different individuals[37] perceive in a similar, although not identical, manner. In particular, the survey reveals that humans are more concordant in conceiving extremely chaotic or ordered textures, whereas a higher ambiguity is present in the transitional levels of the scale.

4.2 The classification algorithm

The specific purpose of this research stage is to obtain a model capable of predicting the averagely human perceived chaos/order level of an audio signal, relying on the collected dataset. The objective is to produce a light and fast enough algorithm to be operated in pseudo-real-time on a common laptop computer. We considered as target a behavior faster than 200 milliseconds on a Macbook Pro 2011[38].

In order to fulfill these requirements we opted for a *Convolutional Neural Network* design, implemented through the Python programming language. This approach has been selected a priori, basing on the information gathered from several papers and researches, some of which are mentioned in the background chapter of this document. In particular:

37 Individuals with a common cultural background.

38 CPU: Intel i5 2,3 GhZ, RAM: 4 Gb 1333 MhZ DDR3.

- ANNs are proved to be an effective choice for fuzzy/ perceptual feature extraction [10];
- CNNs are proved to achieve consistent accuracy in audio classification tasks [27];
- CNNs are proved to be more computationally efficient, compared to competing ANN models (RNNs) [27].

Nevertheless, one significant disadvantage of this technique, compared to a RNN implementation, is that only data with equal dimensionality can be analyzed by the network. In our particular case, this means that only sounds with an exact fixed duration (defined a priori) can be classified.

4.2.1 Introduction to Convolutional Neural Networks

The application in which CNNs excel (but not limited to) is automatic classification of image-related data [45], in fact, their functioning is inspired by the behavior of the visual cortex. As *David H. Hubel* observed in several experiments, for example *Transformation of Information in the Cat's Visual System* [46], certain "neuronal cells", to use his own words, are able to recognize and react to complex aggregation of stimuli. The latter can be thought as *visual patterns* that define archetypical constructs such as "vertical lines" or "curved segments". This is perfectly coherent with the grouping processes identified by the gestalt [7]. Then, the basic concept beyond the CNN is to algorithmically replicate this visual aggregation process, in order to recognize "archetypical" shapes in images and define complex structures (upon these basic shapes) to detect manifold information in data. For example, a CNN task could be establishing if an image portrays a monkey, and determining the particular species of the animal.

An introduction of the basic functioning of a CNN will follow, although it should be considered as a qualitative description, targeted to contextualize the architecture we implemented. It should not be intended as an exhaustive explanation of this deep learning model. All the following information can be verified and further deepened consulting *J. Wu* [45], *Goodfellow et al.* [12], *D. Stutz* [24]. For simplicity, we will first analyze the functioning of a *trained* CNN, that is a model that "already knows" how to achieve its task. Then, we will describe how the training process makes this possible.

Considering a trained CNN as a *black box*, its task usually consists of determining if a data-point belongs to a category (or discerning among multiple possible categories), its input consists of a digitized image (or similarly encoded data) and its output is a prediction (or an array of predictions). The input is usually a multidimensional vector (tensor), as an image with *X* rows, *Y* columns and 3 channels (RGB), although differently shaped tensors are similarly managed. In fact, in CNN based audio applications, STFT matrixes are usually adopted as input, which are shaped as (*n_time_frames, n_fft_bins, 2*). Instead, the output can be either a single or multiple value, which represents the computed probability of matching the inquired category (or categories).

The internal architecture of a CNN is subdivided in different processing algorithms connected in a cascade fashion, which are referred as layers. The information propagates from the input to the output of the network, sequentially passing (and being modified) through each layer. The amount, typology, sequence and hyperparameters[39] of the layers constitute the *architecture* of a CNN and, usually, different tasks require different designs. For example, a CNN arranged to perform object mapping can be completely inaccurate for pictorial style recognition. Accordingly, it is important to implement the most suitable architecture for a specific task.

39 Parameters that can be defined only by the programmer. In other words, they cannot be automatically learned by the network.

The most common layer types adopted in CNNs are: *convolutional*, *nonlinear*, *pooling* and *fully connected,* even though several others are possible.

The behavior of convolutional layers, which are the most present in CNNs, is based on the operation of convolution kernels (also referred as filters). A kernel is a tensor considerably smaller than the input image[40]. The contained values approximate the shape of a visual pattern, representing an archetypical visual feature. Figure 10 shows a simple kernel containing a curved feature.

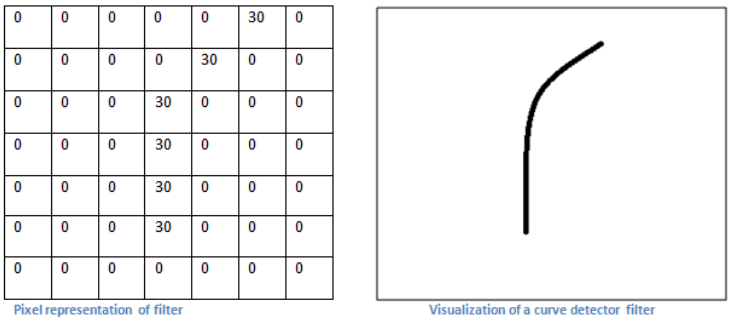

Figure 10: Simple convolution kernel

(*image source: https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/*)

By convolving (point to point multiplication and summation) a kernel with an identically-sized section of the input image, is obtained a value that reflects the amount of matching between the two vectors. This indicates how much the feature carried by the kernel is present in the analyzed image section. The matrix obtained by "striding" the kernel in all possible positions within the input image is called f*eature map*. This tensor represents therefore the presence amount and dislocation of a particular feature (visual pattern) in the whole image. Figure 11 describes the first step of the calculation of a feature map. To obtain the entire feature map, imagine to stride the filter (green matrix), convolving it in all possible

40 Even though the depth size (e.g. 3, for RGB images) is usually the same of the input.

positions of the input image (blue matrix) and put the obtained values in the correspondent positions of the feature map (the matrix on the right).

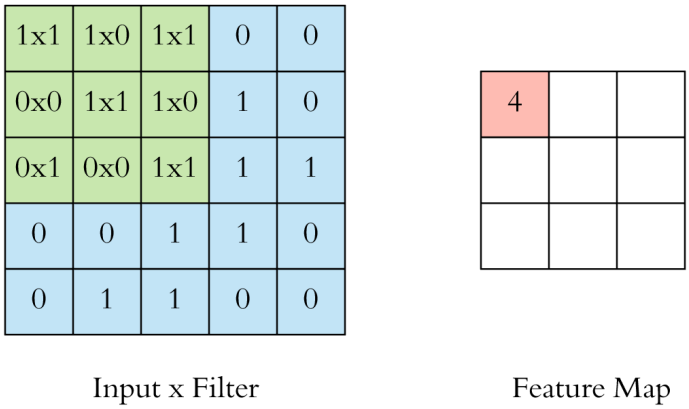

Figure 11: Feature map filled with the first value
(*image source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2*)

Nevertheless, the above-explained scenario is simplistic. In fact, kernels are usually more complex than the one shown in Figure 10. Furthermore, in the real world, a convolutional layer can implement even dozens of them to detect multiple features, constructing high-dimensional feature maps. The amount of kernels present in a convolutional layer defines its *depth* hyperparameter.

Since the operations performed in this kind of layer are strictly *linear* (element-wise multiplication and summation), a common practice is to add *non-linearities*, applying an *activation function* to every element of a feature map. This procedure permits to better match fuzzy features, which involve non-linear correlations among pixels. For example, the species of a portrayed animal certainly can not be mapped through the simple matching of "basic patterns", since it implicates complex and non-linear dependencies dislocated within an image. A frequently adopted non-linear activation function is the *rectified linear unit* (*ReLU)*, which is described by:

$$y = max(0, x)$$

where y is the output of the function and x is its input. This function basically turns all negative values of its input to 0, increasing the nonlinearities of a model and facilitating the learning of fuzzy and complex features. For an exhaustive overview of ReLU refer to *Nair et al.* [47].

A common (but not mandatory) practice in a CNN design is to reduce the dimensionality[41] of the feature maps through dedicated processing algorithms, referred as *pooling* layers (also mentioned as downsampling layers). This process serves mostly to increase the computational efficiency of a model, reducing the amount of calculi to be performed. It is also aimed at minimizing the possibility of overfitting. The most common pooling strategy is called *max pooling*. This technique involves the application of a simple striding kernel, in a similar fashion of convolutional layers. The max pooling filter analyses an image (or feature map) subsection and outputs only its maximum value, populating an output tensor with the down-sampled data, as illustrated by Figure 12.

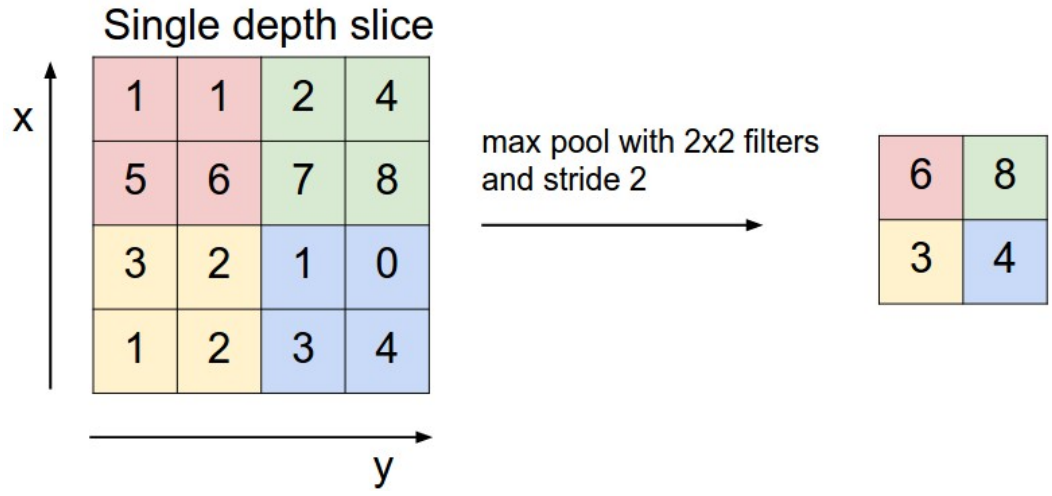

Figure 12: Max Pooling

(*image source: http://cs231n.github.io/convolutional-networks/#fc*)

41 All dimensions except the depth.

After *convolutional, nonlinear and pooling* layers, usually follow at least one *fully connected layer* (although it is possible to concatenate many ones), which is the most commonly adopted in ANNs in general. The function of these layers is to identify which and how much its input elements[42] are correlated to match a particular category. These layers consist of sets of parallel n*eurons,* which are simple computing cells. Neurons contain basic processing structures (mathematical operations applied to their input) referred, again, as *activation functions*. Several types of them are possible (for example: *Sigmoid*, *Binary Step*, *TanH*, *ReLU*) and the choice of the most effective one can sharply influence the accuracy of an ANN model. The amount of neurons defines the *depth* hyperparameter of this kind of layer. In a fully connected, all input elements are connected to every neuron and all connections are rescaled multiplying the input by a *weight* factor[43]. Thus, for every neuron, the output is computed by summing its rescaled inputs and applying the activation function to the obtained value.

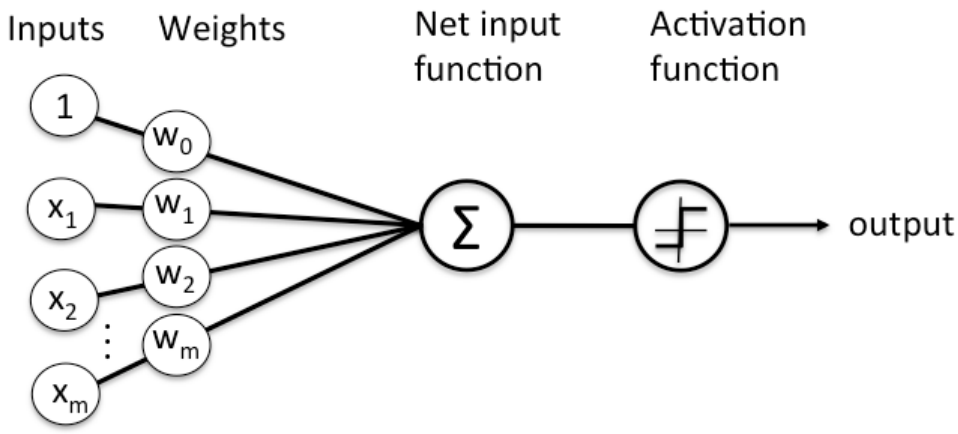

Figure 13: Inputs and output of a single neuron in a fully connected layer (*image source: https://deeplearning4j.org/neuralnet-overview*)

42 Which, in CNNs, are usually the cells of feature maps extracted by convolutional layers and eventually downsampled.

43 In certain circumstances also a *bias* is added.

Figure 13 shows the input/output connections of a single neuron of a fully connected. The final layer of a CNN is always a *fully connected* and must have a *depth* size equal to the number of classes that the model should distinguish. For example, a design targeted to discriminate if an image portraits "happy" or "sad" people, should present a final fully connected layer with only 2 neurons (with *depth* equal to 2). The activation function of the neurons in the last layer is usually a function that permits to obtain an output array of predictions, in which every value is interpretable as a probability (of matching one category). For this purpose it is common to adopt a *softmax* function, which causes the network outputting a series of values in the range (0,1), which sum up to 1.

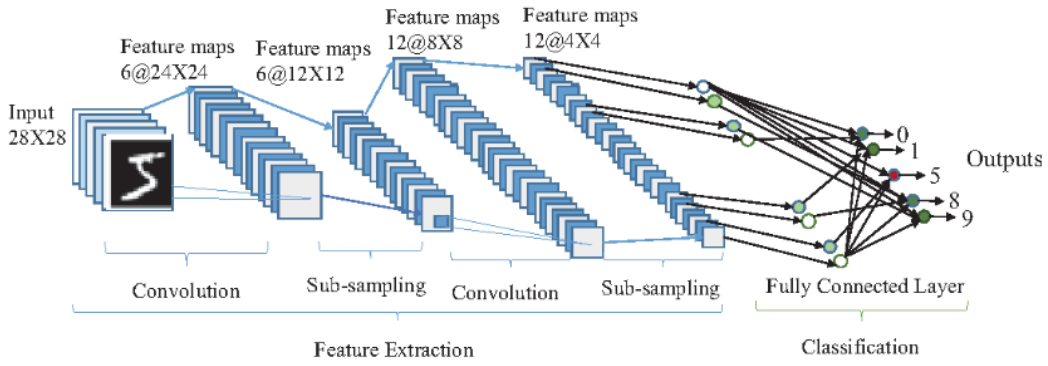

Figure 14: Block diagram of a typical Convolutional Neural Network (image source: https://www.semanticscholar.org/paper/Memristor-crossbar-deep-network-implementation-on-a-Yakopcic-Alom/59d7c6c1e0b761aed209b2acbc2a181db29860c7)

A common CNN designs involves a concatenation of several layers in a cascade fashion. Figure 14 shows an example of CNN aimed at the classification of handwritten digits. Following this architecture from the input to the output, the original data becomes represented by increasingly higher level features (more and more fuzzy, abstract and nonlinear). This occurs because, going through the network (forward propagation), the data represented in the layers become function of the previously extracted

features, defining more and more complex superstructures. Therefore, the concatenation of *convolutional/nonlinear/pooling* layers serves only to extract features from the input data, which represent data characteristics that are meaningful for the network's task (and are often incomprehensible for humans). The final representation of these features (the last feature map) is then flattened[44] and propagated trough the fully connected layers. In this part of the network occurs the actual classification process, identifying the statistical proportion among the extracted features, which determines a data-point to belong to a category. The prediction is finally computed in the last fully connected layer, outputting the probabilities of matching the inquired class(es).

As stated above, the behavior described so far is referred to a *trained* network. This means that its architecture and parameters are correctly set to perform its task. While the architecture and hyperparameters are defined a priori for a network, its *parameters* are initially unknown. The latter consist of the data contained in the kernels of the convolutional layers and the weights applied to every connection of the fully connected ones. The training process of an ANN is aimed at automatically identifying the network's *parameters* that permit a correct classification of the desired categories. In the case of a classic CNN design, this means to find meaningful kernels, and thus, features that are useful for the network's task and the appropriate weights that determine how the features are correlated for every inquired data category. On the contrary, the hyper*parameters* relative the network design (layers sequence, depth ecc.) can not be learned by the network on its own and must be carefully selected by the programmer. We recall that, in supervised learning problems the training process relies on labelled datasets, in which every data-point is associated to a numerical value that expresses which class it belongs to[45].

44 Turned from a multidimensional shape to a 1D array.

45 For example, "happy people" could be 0 and "sad people" could be 1.

In the training process, all parameters are randomly initialized and then they are fine-tuned through an optimization process that is divided in consecutive stages. Initially, a data-point of the training dataset forward propagates through the network, generating a prediction. The latter is potentially wrong at the beginning, since the network parameters are still random or non optimized. Then, a *loss function* of the prediction is computed, comparing the latter with the expected outcome (the label). This value represents the current model's accuracy, in other words, how much an obtained prediction is distant from the truth. Several loss functions can be adopted and the choice of this hyperparameter influences various aspects of the training, among which, the training speed and the obtainable accuracy. A common loss function for CNN-based categorization applications is *categorical crossentropy*. Successively, through a backpropagation process, every single parameter of the network is slightly altered towards a direction that is expected to reduce the loss function, and thus to produce more accurate predictions. The computation of the direction and amount of the parameters' alteration is entrusted to a *gradient descent* algorithm and is influenced by the *learning rate*. The latter is a simple weight factor applied to the values computed by the gradient descent, thus it rescales the update range of the parameters, defining a maximum excursion. This hypermarameter can influence the final accuracy, as well as the amount of updates necessary to reach the maximum accuracy. This process is performed for all data-points, eventually grouped in *batches,* updating the networks's parameters only after the forward/backward pass of all data contained in a batch.

The training usually requires various "cycles", or *epochs,* to reach a reasonable accuracy. A training epoch occurs when all available data-points are passed through the forward-backward propagation process. Therefore, the set of network's parameters obtained after every epoch can be considered a model on its own. Nevertheless, at the end of the training

process, only one parameters-set is chosen as definitive model and usually is the one that provides the best accuracy[46]. The *batch size* is an important hyperparameter, which can influence the final accuracy, computation speed and memory required for the training.

The objective of a CNN (and ANN) design and training is thus to obtain a single set of network's parameters and hyperparameters to obtain the most accurate predictions for any data similar to the training data-points. A correctly implemented CNN is therefore capable of generalizing its task, performing accurate predictions on data unobserved during the training stage.

4.2.2 The implemented analysis architecture

The prediction accuracy of a CNN, and of ANNs in general, is sharply influenced by the representation typology of its input data. In fact, as explained above, a prior extraction of *motivated* (task-related) features can increase the accuracy of a model, focusing the training on relevant aspects of the input data.

In the case of audio classification, spectrograms (STFT) are proved to provide more accurate outcomes, compared to feeding CNNs with time-domain waveforms [22]. Then, the collected data has been preprocessed in order to obtain an appropriate spectral representation of the sound textures. In particular, we adopted 1024-samples non-overlapped hamming windows and a 1024-sample FFT applied to every window. The choice of non overlapping frames is aimed at reducing the computing requirements. To further decrease the amount of calculi, the spectral phase information has been discarded, computing the absolute value of the Fourier transform. To

46 The best model could not be the one correspondent to the last epoch of the training.

perform these operations we employed Python functions extracted from the above-mentioned SMS-Tools library[47]. The dataset has been arranged into 2 independent tensors. The first is the *predictors* matrix, containing the STFTs of all data-points, shaped as (n_data, n_frames, n_bins), where the first dimension is the total number of data-points, the second is the amount of FFT frames (time) and the third is the amount of FFT bins of every frame (frequencies). The other is the *target* matrix and contains the one-hot-encoded[48] human classifications. It is shaped as (n_data, n_classes), where the first dimension is, again, the total number of data-points and the second is the amount of possible order classes.

We experimented various CNN architectures and hyperparameters settings, although, as stated at the very beginning of this document, we will report only the final implemented design. The latter has been selected according to the highest obtained accuracy. The architecture we realized has been inspired by the work of *Salomon et al.* [48], which developed a CNN-based algorithm for the classification of environmental sounds. The technical realization of our network is based on the *Keras* [49] library, which consists of a *Tensorflow's* [50] *API.* The implemented design is summarized in the block diagram portrayed in Figure 15. The graph shows the data flow inside the network from its input (top) to the output (bottom). To the left of the chart is represented the sequence of the implemented layers, each correlated, on its right, with its selected hyperparameters. The feature extraction block of this design consists of the first 9 layers (from the

47 https://github.com/MTG/sms-tools

48 One-hot-encoding consists of transforming a series of scalars into one-dimensional vectors with as much elements as the maximum value present in the series. In particular, the output vectors contain only one "1", located at the position expressed by the input scalar and all other values are 0. For example, in our case, the series range is 11 (order classes). Accordingly, a sound classified as "5" becomes [0 0 0 0 1 0 0 0 0 0 0]. This representation is coherent with the output prediction vector of the CNN model.

first to the last 2D Convolution), while the classification task is accomplished through the last 2 concatenated fully connected.

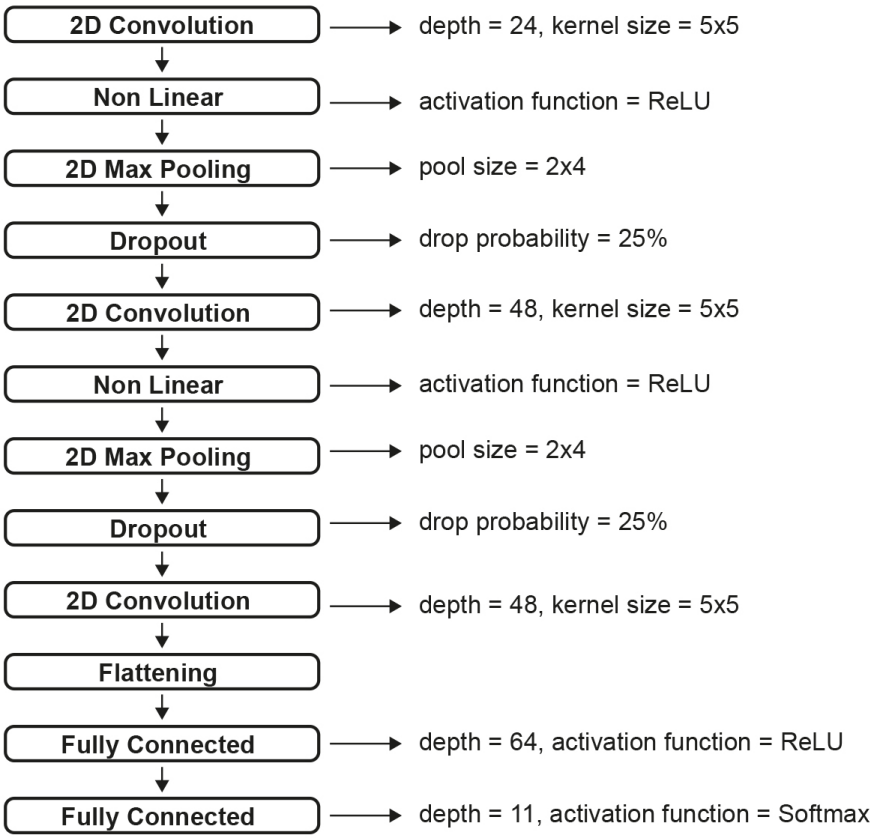

Figure 15: Implemented Convolutional Neural Network architecture

The adopted kernel dimensions for the convolutional layers (5x5) indicates that the extracted features are included in spectral regions of approximately 23 milliseconds x 172 Hz. The pooling size has been selected following an empirical intuition based on the adjectives collected through the classification test. In fact, as explained above, several reported attributes (for example *crumbled, discontinuous*, *static*, *constant*...) suggest that the majority of testers adopted time-variant perceptual structures for discriminating chaotic and ordered textures, rather than static proportions among frequency-related information. For this reason, the implemented size of the pooling filters is 2x4, respectively referring to the time and frequency

axes. This makes the downsampling more effective for the spectral (static) information, maintaining a softer approximation for the time-related dependencies. The use of *dropout* layers serves at reducing the possibility of overfitting. These layers momentary deactivate a given percentage (25%) of random input connections[49], forcing the model to not rely on particular features to learn a concept [51]. The depth size (of convolutional and fully connected[50]) and the activation functions (of non linear and fully connected) have been maintained coherent with the values implemented in the experiment of *Salomon et al.* [48] since, after several experimentations, they provided the best accuracy for this architecture. To define the amount of training epochs we adopted a technique called *early stopping,* which consists of interrupting the training process when the accuracy does not improve any more performing new training cycles. This can prevent overfitting issues, avoiding that the network's parameters overly adapt to the training data. The training has been performed with a batch size of 1 data-point, adopting *Categorical Crossentropy* as loss function and a learning rate optimized by an ADAM algorithm [52].

4.2.3 Automatic classification accuracy

We report the results obtained through 4 separate trainings, performed adopting 4 distinct datasets:

- Original dataset, containing only the samples selected after the dataset human classification stage. It includes 297 data-points, consisting of 3-seconds labelled samples;

49 In the case of a pooling or a convolutional layer as input, cells of the feature map are deactivated.

50 Obviously, the last fully connected has a depth size equal to the classes to be predicted in this context (11).

- Augmented dataset, containing the samples of the original dataset, as well as the ones obtained through the only augmentation procedure (producing 10 alternative versions for each data-point). It counts 3267 data-points, which are, again, 3-seconds labelled audio files;
- Segmented dataset, which includes the sounds achieved processing the original dataset through the only segmentation algorithm. It contains 3267 data-points, consisting of 0.5-seconds labelled samples;
- Augmented and segmented dataset, which contains the sounds obtained passing the augmented dataset through the segmentation algorithm. It incorporates a total of 35760 data-points, which consist of 0.5-seconds labelled files.

The trainings have been executed adopting the same hyperparameters, in order to be able to directly compare the effectiveness of the implemented dataset augmentation and segmentation. To correctly perform the training process and assess the models' accuracy, we split every dataset in 3 sub-datasets:

- Training (80% of the dataset). This data is adopted by the network to update its *parameters* during the training process;
- Validation (10% of the dataset). This data is adopted to test the network's performance on unobserved data, and therefore its ability to generalize the learned concepts. We considered the validation accuracy (assessed evaluating the models with the validation set) to affine the *hyperparameters* settings during the development stage of the CNN architecture;
- Test (10% of the dataset). This data is adopted to assess the models' performance with data that has not been used neither for the parameters', nor for the hyperparameters' adjustment. The accuracy obtained evaluating the model with this data is therefore the most reliable measure of a model's performance.

In the following diagrams, the X axis reports the training epochs' progression, while in the Y axis are plotted the training accuracy (in blue) and the validation accuracy (in orange). The test accuracy has been assessed at the end of every training, evaluating only the best achieved model according to the validation accuracy, therefore it is not printed in the graphs.

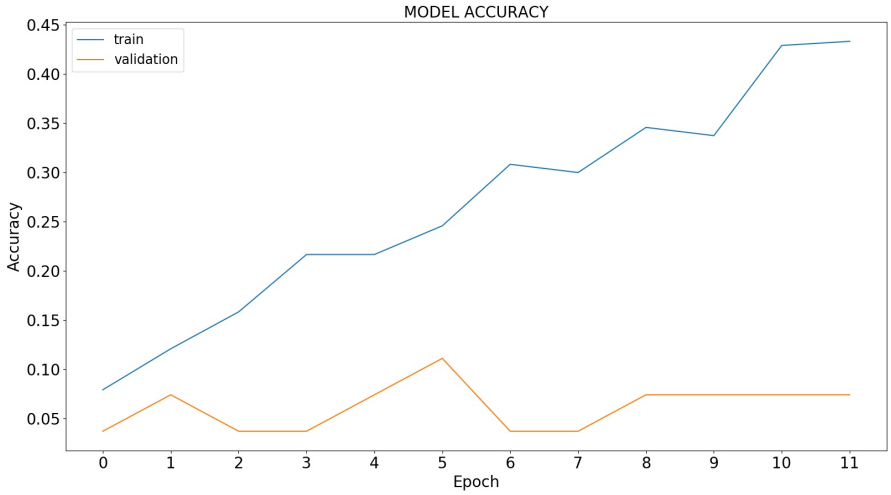

Figure 16: Accuracy of the CNN trained with the original dataset

Figure 16 shows the obtained accuracy for the dataset without neither augmentation, nor segmentation. The training stopped after 12 epochs. Such a short training (interrupted by the early stopping algorithm) indicates that the dataset contains insufficient (or not optimized) information to permit the model to generalize its task to unobserved data (validation set). In fact, the precision is evidently poor, reaching 43.3% for the training samples, 11.1% for the validation set, and a 10.6% assessed through the test data. These outcomes clearly indicate that the model strongly overfitted. In fact, a significantly higher training precision suggests that the model achieved a certain (although poor) comprehension about how to classify the observed data, but it is unable to properly generalize the learned concepts to new inputs. Furthermore, the validation accuracy clearly fluctuates around a

fixed value and do not present an increasing tendency. Accordingly, the selected CNN architecture trained with the original dataset is certainly not a reliable system to properly classify the inquired feature.

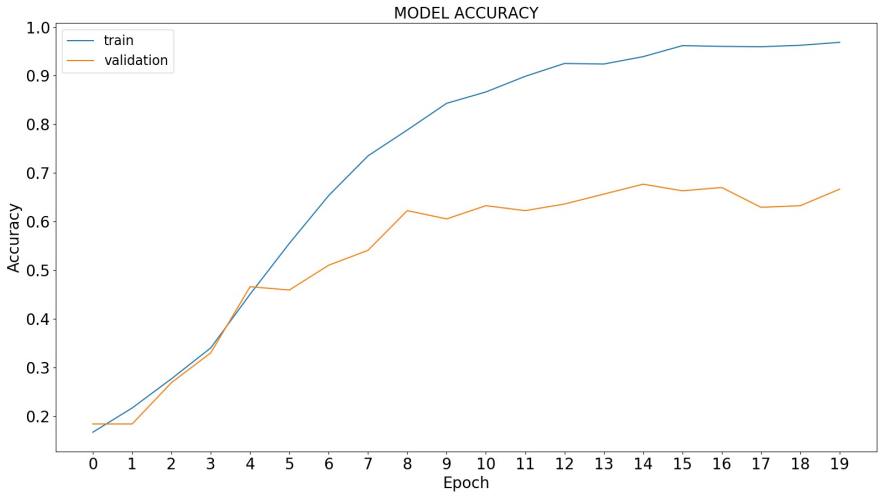

Figure 17: Accuracy of the CNN trained with the augmented dataset

Figure 17 represents the obtained accuracy after the only augmentation process. The training stopped at epoch 20, indicating an increment of useful information in the dataset. It is evident a significantly higher accuracy, which reached 96.8% for the training set, 67.7% for the validation one, and 66.1% for the test data. This reveals that the augmentation procedure conferred a substantial improvement to the model and that the latter, provided with an enough large dataset, could be able to properly solve its task. Nevertheless, a strong overfitting is still present, considering the consistent distance between the training and validation/test performances.

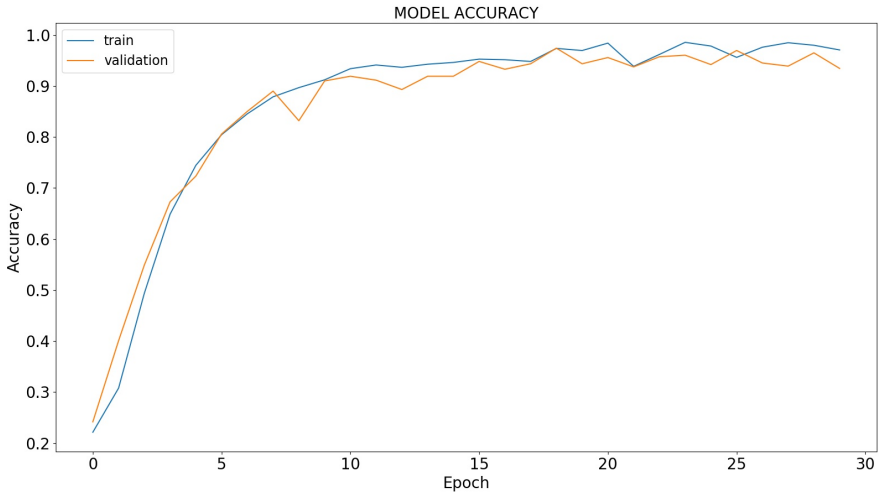

Figure 18: Accuracy of the CNN trained with the segmented dataset

Figure 18 represents the obtained accuracy processing the sounds through the only segmentation algorithm. Here the process stopped at epoch 30, demonstrating a significant optimization of the utile information in the dataset. While the training precision is slightly superior, but comparable to the one achieved by the only augmentation (98.5%), the validation and test accuracies are significantly higher, reaching respectively 97.4% and 96.3%. Moreover, the overfitting has been substantially reduced, presenting very close training and validation/test performances. Therefore, with en equal amount of data-points (3267), the segmentation provides consistently better results, compared to the only augmentation. This is due to the fact that a reduction of the input data dimensionality causes a drastic decrement of the network's parameters to be computed. This permits a considerable simplification and optimization of the training process.

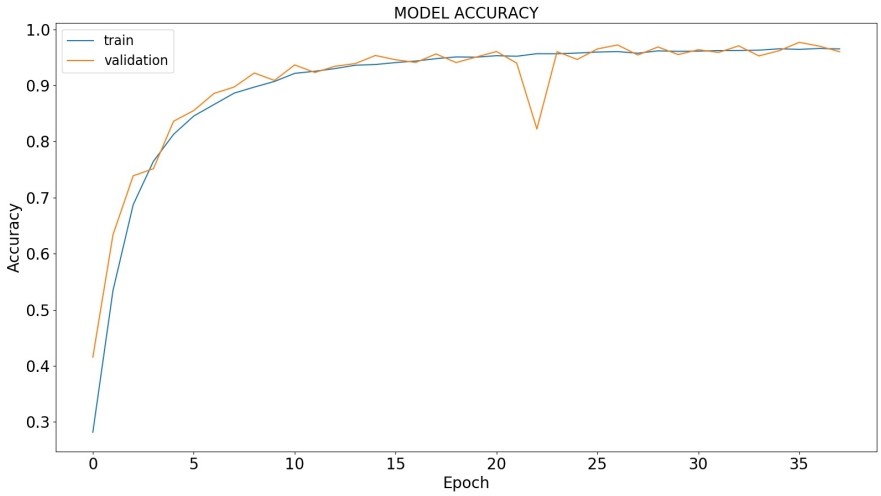

Figure 19: Accuracy of the CNN trained with the augmented and segmented dataset

Figure 19 displays the achieved accuracy for the augmented and segmented dataset. The training stopped after 38 epochs in this case, pointing out a further improvement in the dataset's effectiveness. These outcomes are evidently not distant from the ones obtained with the only segmented dataset. In particular, we reached a training accuracy of 96.7% a validation precision of 97.7%, whereas the test accuracy we assessed is equal to 96.9%. While the training accuracy is slightly inferior than the one achieved with the segmented dataset, the validation and test ones are marginally superior. Standing on what can be inferred from Figure 19, the combination of the developed augmentation and segmentation techniques eliminated any overfitting issue. Accordingly, we consider the model trained with the augmented and segmented dataset as a reliable tool to perform automatic classification of the human perceived chaos/order level in sound information. Nevertheless, it is important to state that the obtained accuracy is referred to our specific dataset. In fact, it is not proved that the model would present an identical accuracy for different sound typologies.

Finally, the network has been arranged for a pseudo real-time utilization, in order to continuously classify audio streams coming, for instance, from microphones. For this purpose, we implemented a "striding" buffer, with a fixed size equal to the duration of the trained sounds (0.5 seconds). The buffer is cyclically updated with samples coming directly from the sound card[51] and, on every cycle, the spectrogram of the buffer's content is computed, as performed for the training data-points. These procedure ensures that the incoming information presents a shape identical to the trained data. The spectrogram is then forward propagated through the CNN generating a prediction about its perceptual order level. This permits to have a constant classification of an audio stream, basing on the last 0.5 seconds recorded.

The system latency in our laptop is estimated at around 100 milliseconds, which can be considered enough small for a fluent and reactive utilization. We empirically compared the performance of the 2 best achieved models (segmented and augmented + segmented), standing on our personal perception of the inquired feature. The augmented and segmented model showed an overall better performance, especially for timbres captured through microphones[52]. As expected, the implemented augmentation technique (in particular the random equalization and random convolution with room IRs) improved the model's generalization ability for real recorded sounds. Accordingly, this model has been selected as the definitive automatic signal classification engine for this project.

At the end of the second stage of our research, we obtained a Convolutional Neural Network model capable of predicting in pseudo-real-time the human perceived chaos/order level of an audio stream, providing

51 By steps of 512 samples.

52 The performance has been assessed both trough dynamic (Shure SM57) and condenser (Behringer C2) microphones, obtaining very similar behaviors.

an accuracy of 96.7% for our dataset, assessed evaluating the model on test data (unobserved neither for parameters', nor for hyperparameters' adjustment). The model has been trained with the collected dataset processed through the augmentation and segmentation algorithms. The final dataset size reached a total of 35760 0.5-seconds sound samples, labelled with their averagely perceived order level. When tested with real-world sounds, captured by microphones, the model shows an accurate performance, demonstrating a satisfactory ability of generalizing the learned concepts.

4.3 The re-synthesis algorithm

This research stage is aimed at obtaining an algorithm capable of producing audible sound textures that present a desired level of perceptual chaos/order. Also here, the objective is to produce a light and fast enough algorithm to be operated in pseudo-real-time on a common laptop computer. To obtain this, we have sought a compromise between computing efficiency, algorithm effectiveness and development agility.

The implemented re-synthesis section is divided in 2 distinct algorithms: a granular synthesizer, which creates sound textures by processing existing audio files, and a Markov-chain-based parameters-sets generator, which controls the behavior of the granular synthesizer. The first has been entirely implemented in Max Msp, instead the second has been coded in Python. The principal reasons that support this architecture choice are:

- Markov chains are proved to be effective for feature-matching data synthesis tasks, as demonstrate various HMM-based implementations aimed at speech synthesis [29];

- For tasks that do not involve large training datasets, Markov chains are proved to reach a reasonable accuracy, comparable to ANN-based competing models [36];
- Markov chains are proved to be more computationally efficient, compared to ANN-based competing models [36];
- Granular synthesis permits to conveniently maintain our study in the domain of the sound texture;
- Granular synthesis permits to efficiently obtain all considered chaos/order classes through a restricted set of parameters.

A strong limitation that this architecture imposes is that granular synthesis is not a "generalizable" model, as could be considered, for example, additive synthesis or Fourier-based techniques. In fact, through the latter is possible to virtually generate any kind of sound, property that can difficultly be achieved through granular synthesis. Therefore, we are conscious that this approach makes our re-synthesis algorithm non generalizable to any possible sound archetype. Nevertheless, at this stage, we considered this method as the most efficient stratagem to obtain a properly-working archetypical re-synthesis architecture. However, we created the algorithm with a *modular* conception, predisposing the substitution of the granular synthesizer with any other synthesis model by applying restricted modifications to the code.

The implemented granular synthesizer is strictly similar to the one adopted for the dataset creation. The principal difference is that the re-synthesis granular algorithm processes only *one* single large audio buffer, instead of 100 smaller ones. This modification is due the fact that Max Msp introduces a significant latency when loading buffers and, for real time operations, a single pre-loaded file provides a faster response. The large audio sample has been built by concatenating portions of the same sounds (downloaded from the Freesound database) adopted to create the

classification dataset and presents a total duration of 42.18 minutes[53]. The parameters-set needed by the granular synthesizer consists of an array of 8 values:

1. Metronome speed;
2. Initial slice position;
3. Position randomness;
4. Initial slice duration;
5. Slice duration randomness;
6. Initial playback speed;
7. Playback speed randomness;
8. Envelope type.

The Source Sample ID parameter is omitted here because of the adoption of the single buffer. The parameters' arrays are generated by a simple Markov process, capable of selecting the correct values to synthesize a texture with a desired perceptual order level.

4.3.1 Re-synthesis dataset creation

Since Markov chains are experience-based processes (as ANNs), to obtain the target behavior it has been necessary to collect another dataset, containing parameters-sets associated with the perceptual order level of the correspondent generated textures. This can be thought as a database of labelled *presets* for the granular synthesizer. This dataset consists of the experience of the Markov chain, from which learn how to produce new semi-aleatory presets that match a desired order level. To avoid any possible confusion, from here onwards we will refer to the dataset collected for the CNN training as *classification dataset* and to the one aimed at the

53 This operation has been performed through the software Pro Tools HD 10.

synthesis parameters-sets generation as *re-synthesis dataset*. The latter has been generated through an automated algorithm and, again, it has been built in order to obtain a homogeneous balance among its classes. A block diagram of the algorithm implemented for the re-synthesis dataset creation is represented in Figure 20.

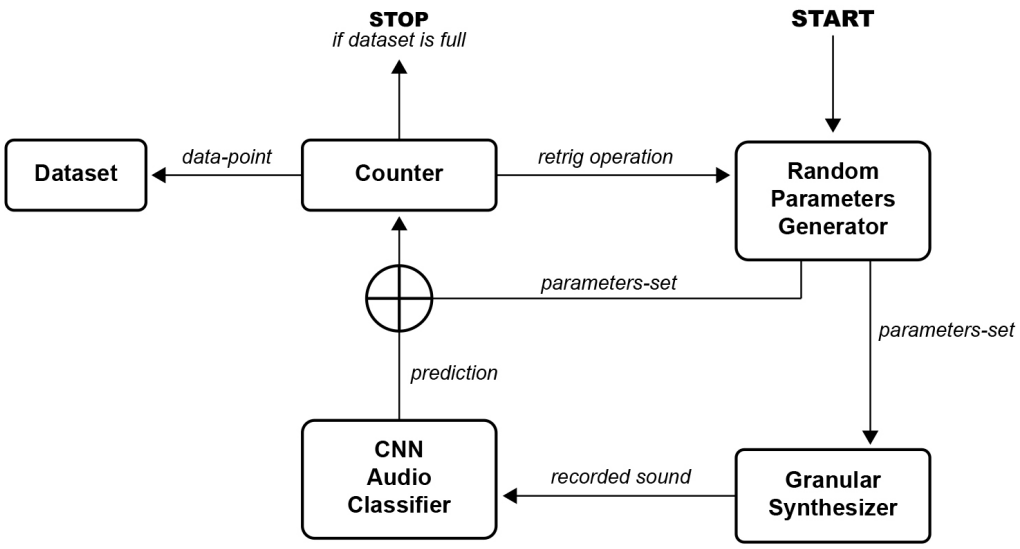

Figure 20: Block diagram of the re-synthesis dataset creation algorithm

To obtain one single data-point (one parameters-set labelled with its correspondent order level), first a random array is created. This consists of 8 random values (as much as the synthesis parameters), individually restricted within the utile range of the target parameter. Successively, a texture is generated by tuning the granular synthesizer with the random values and the obtained sound is recorded in a 3-seconds buffer[54]. The buffer is then segmented in 0.5-seconds overlapped frames and the STFT is computed for every slice, exactly as occurred in the segmentation stage of the classification dataset. Every frame is consequently forward propagated through the CNN, in order to predict its perceptual order level. The rounded

54 Since the random presets occasionally produce textures that are practically silence, sounds that present a RMS level below 0.05 are automatically discarded. This procedure has not been necessary in the classification dataset creation, because every data-point was selected by a real person.

mean of the predictions (of every frame of a 3-seconds recorded texture) individuates then the label to be associated with a generated parameters-set. Finally, the synthesis parameters and the corresponding label are queued in two distinct tensors, as occurred for the predictors and target matrixes of the classification dataset.

To obtain a homogeneous collection, an equal amount of data-points for each class (11 in total) has been collected, reaching a 2200-points dataset, with 200 data-points for every class. Due to the error rate of the automatic classification algorithm, we manually discarded approximately 40 data-points, which were evidently classified with the wrong class (according to our perception). Since the random generation sharply tends to produce textures unbalanced towards the perceptual chaos, as explained above, the implemented automatic re-synthesis dataset population algorithm requires relatively long times. In fact, the creation of 2200 equally distributed data-points took approximately 8 hours of computing. Beyond this inconvenient, which is essentially related to the specific architecture of our granular synthesizer, this method provides the possibility of building multiple re-synthesis datasets. For example, adopting a different audio buffer for the granular synthesizer would permit to obtain diverse sonic results. Furthermore, by selecting the appropriate size and range for the parameters to be generated, it is possible to apply the same procedure to different audio synthesis algorithms, for instance FM or additive. Thus, this technique permits to automatically create preset databases for any kind of synthesis model, associating to every preset its estimated perceptual order level. Furthermore, by substituting the CNN model, it would be possible to create datasets catalogued according to any other sound archetype(s).

4.3.2 The implemented re-synthesis architecture

As explained above, a Markov chain is an aleatory process aimed at generating plausible sequences, by imitating existing examples belonging to the same complexity. For an exhaustive explanation of Markov chains, please refer to *A. Tolver* [53].

In this particular instance, the given examples, that are the experience of the Markov chain, consist of all the re-synthesis parameters-sets, matched with their perceptual order level label. Since different parameters-sets can produce textures associable with the same perceptual class, a probabilistic proportion among the single parameters can define a synthesis model of that class. A parameters-set can be intended as a sequence of states, in which every state is a single parameter. A *transition matrix* defines the probability of passing from a *state* to the successive one, identifying the proportions between adjacent parameters that statistically produce a texture with a certain order level. In this circumstance, the probability of transition from one state to the successive indicates the correlation level between two parameters that are adjacent in the sequence. For instance, how much is probable that, to produce a certain order class, a 30 milliseconds *metronome speed* (that is the first value of the parameters-set) could be associated to a 12 seconds *initial slice position* (that is the second value). The definition of a transition matrix for every possible order class provides distinct models of "how the granular synthesizer should be set to produce textures matching a precise class". Thus, selecting a specific order class to be synthesized, corresponds to choose its relative transition matrix. This concept is strictly correlated to the idea of "screen", introduced by *Iannis Xenakis* for his composition "*Analogique"* [54].

To obtain this behavior, the re-synthesis dataset has been split in 11 separated sub-datasets, each containing all the data-points correlated to one

perceptual order class. For each sub-dataset, every position of the sequence is associated to a dictionary containing all its candidate values. Then, to produce a new sequence, an aleatory process progressively selects a random candidate for each state, generating a new parameters-set for the granular synthesizer. The candidate values for one state strictly depend on the previous adjacent state that is chosen. In particular, a value B can succeed a value A if and only if the sub-dataset (the experience) contains the sequence AB, in the relative sequential positions. Furthermore, the probability that B is chosen after A depends on how much times the sequence AB appears in the sub-dataset, always in the relative positions. The more times 2 adjacent values appear in the dataset, the more probable is that they would appear also in the generated sequences. This arrangement implicates that the system is *memoryless,* satisfying the *Markov property.* This means that a particular state is dependent only by the immediately precedent one, and not by all others. For example, a value of 12 for the *initial slice position* could succeed a value of 30 for the *metronome speed* if and only if:

- 30 is randomly chosen as first state;
- In the sub-dataset, at least one data-point contains a 30 metronome speed and a 12 initial slice position.

Furthermore, the more presets in the dataset contain a 30 metronome speed and a 12 initial slice position, the more probable is that 12 is chosen after 30. Since the first state does not have “previous states”, it is randomly selected among the all present in the sub-dataset. Successively, the next parameters are procedurally chosen according to the previous adjacent states in a “chain” fashion (1=>2, 2=>3, 3=>4, 4=>5, 5=>6, 6=>7, 7=>8). Therefore, when an order class is selected, the Markov chain reconstructs a sequence following the relative transition matrix and generates a parameters-set matching the desired category. The probability $Px_{n,n-1}$ that a value $x_{n,n-1}$ is chosen in the position n of the sequence, after a value x_{n-1} in the immediately precedent position ($n-1$) is given by:

$$Px_{n,n-1} = \frac{N_n}{N_{n-1}} \cdot 100 \qquad N_{n-1} = \begin{cases} N_{n-1} & if \ n>0 \\ L & if \ n=0 \end{cases}$$

where: n is the considered position in the sequence. N_n is the amount of times a value x_n appears in the dataset at the position *n* of a sequence, immediately after a value x_{n-1} at the position *n-1*. N_{n-1} is the amount of times a value x_{n-1} appears in the dataset at the position *n-1* of a sequence. L is the total amount of data-points present in the dataset.

Figure 21 shows a transition matrix similar to the ones adopted in our re-synthesis algorithm. This simplistic example considers the following sequences of letters as dataset:

- [A, D, F];
- [A, E, G];
- [B, E, G];
- [C, E, H].

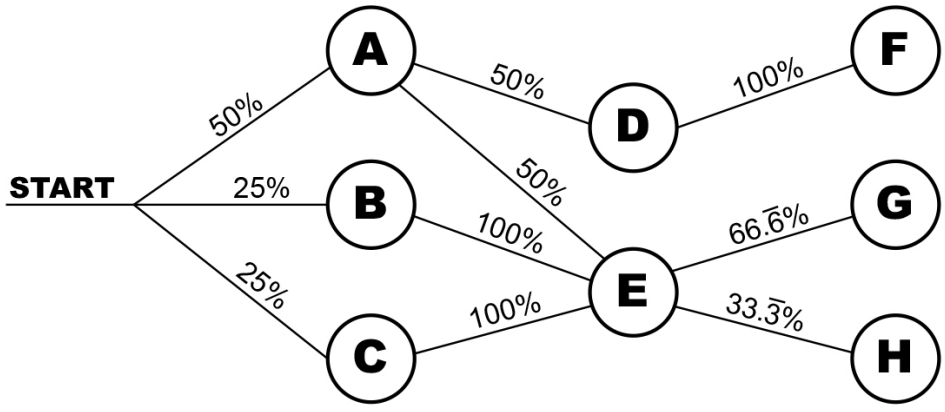

Figure 21: Graphical representation of a simple transition matrix

The Markov chain goes through this diagram from the left to the right, randomly selecting the path to follow and always producing a 3-states sequence. The percentage levels in the connections represent the probability

that a certain state is selected. The connection presenting a probability of 0% are omitted in this graph. The amount of possible sequences is obviously scarce in this case, being the dataset limited to 4 3-states data-points. Although, the diagram shows that it is possible to construct sequences that are not present in the dataset, for example [C, E, G]. While the chain dependencies ensure that the sequences produce textures matching a desired order class, the semi-random selection permits to obtain a volume of different parameters-sets considerably higher than the amount of presets contained in the dataset. Although, this semi-random preset generation process is only capable of concatenating sub-sequences of its experience, that are values present also in the dataset. In fact, it is not able to produce completely new parameter-sets (as can be evinced also by Figure 21). This Markov chain algorithm can be adopted to generate the parameters-sets needed by any other synthesis algorithm. In fact, by simply setting a different number of states to compute, it adapts to any similarly-shaped re-synthesis dataset.

Finally, the parameters-set generator has been connected with the granular synthesizer through the O*pen Sound Control* (OSC) protocol[55], in order to be able to request and instantly hear audible textures presenting a desired order class. Since the computing of the Markov chain is very fast, this operation is perfectly performed in real-time, without any audible latency (< 1 millisecond).

4.3.3 Re-synthesis accuracy

Once implemented the re-synthesis algorithm, a survey has been proposed to humans to verify the perceptual accuracy of the generated

55 http://opensoundcontrol.org

textures. To avoid any possible confusion, we will refer to this test as *re-synthesis survey* and to the previously-described one as *dataset-classification survey*. The implemented re-synthesis test is practically identical to the precedent one. The major differences are the sequent:

- Every test comprehends a total of 33 timbres to be classified (instead of 50), 3 for each order class;
- All proposed timbres are generated through the re-synthesis algorithm (instead of being selected within the classification dataset);
- The test does not request to list possible adjectives correlated with chaotic and ordered timbres.

Unfortunately, due to mere lack of time, it has not been possible to involve a considerable amount of testers, as occurred for the previous survey. In this place, a total of 10 persons performed the test, which led to a collection of 330 classifications, divided in 30 judgements for every possible order class generated by the re-synthesis algorithm. Every instance has been proposed in a relatively quiet room. Every subject adopted the same headphones (Beyerdynamic DT 770 PRO) and the same laptop (Macbook Pro 2011). The test consists of 2 consecutive sections:

1. General and attitudinal questions;
2. Sounds classification.

Every section is provided with a clear explanation of the tasks to be accomplished, as well as instructions for the interface's usage. For each instance, the following general and attitudinal questions are initially asked, which are identical to the ones found in the dataset-classification survey:

- Name and surname initials;
- Age;
- Have you ever had hearing dysfunctions?;
- Have you ever studied in depth sound or music related subjects?;

- Do you regularly practice sound or music related activities?

These information served to identify the person who took a particular test, to eventually re-propose it in case of loss of data. Since the previous test revealed that the bias introduced by the age and the musical practice/education of the subjects is restricted, in this instance we equally weighted all data right away.

Before the actual sound classification, the tester listens to series of 5 example sounds, generated by the re-synthesis algorithm. The requested order class is randomly chosen for every example. Exactly like for the previous survey, this procedure serves to give a preventive idea of the timbres to classify, reducing the possibility of biasing the judgements of the first sounds. After this stage, the subject has to classify 33 distinct textures and the proposed sequence is randomly generated to prevent possible influence in the judgements. Also here, the classification is organized as a Likert-type ascendent scale, visually proposed as a series of 11 check boxes (of which only one can be selected). The boxes' numbering is omitted, giving instead qualitative descriptions positioned at the extremes and center of the scale (*maximum chaos, balance, maximum order)*. Moreover, the scale direction is randomized for every test, as occurred in the dataset-classification survey. The visual layout is identical to the one found in the previous test, arranged to minimize possible biases derived from non optimal displacement of the objects. Finally, after the classification of all sounds, the user is requested to review all answers from the beginning, in order to prevent eventual biases in the first classifications. As occurred for the dataset-classification survey, in the analysis of the obtained results we will refer only to a *chaos-to-order* scale, going from 0 (chaos) to 10 (order) and all classification given for the inverse range have been opportunely rescaled to match this univocal representation.

We collected a total amount of 327 (over 330) classifications, whereas, for 3 sounds, an abstention of judgement is reported. All subjects belong to

the same cultural background, being italian and resident in Veneto. The age range is spread within 23 and 57, with a mean age equal to 34.5. Only 1 tester presents auditory dysfunctions (tinnitus). 3 subjects assert to not have a musical background, neither regarding music or sound related studies, nor concerning practical activities. On the contrary, 7 persons have a sound/musical background, among which, 4 regarding studies and practice, 2 concerning the only practice and only 1 referred just to the studies.

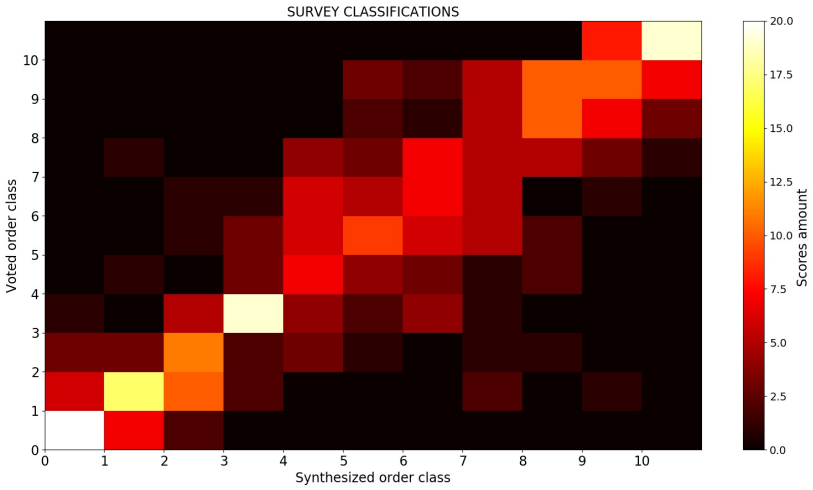

Figure 22: Collected classifications for each order class

Figure 22 represents on the X axis the order class requested to the algorithm, on the Y axis the classifications given by the testers and the Z axis (color brightness) refers to the amount of votes collected in each point. A certain coherence between the expected behavior and the human perception in evident, being the scores concentrated in the diagonal line going from the bottom-left to the upper-right of the graph. A majority of points in the bottom-right side indicates that the algorithm generally tends to produce textures slightly unbalanced towards the perceptual chaos. Furthermore, as we expected, a minor ambiguity is clear for the classes at the extremes of the scale (0 and 10). Instead, the transitional levels present a higher spread of judgements with an apparent peak of dispersion on class 7.

This reflects the previously-identified class-wise ambiguity level of the human perception about the inquired feature.

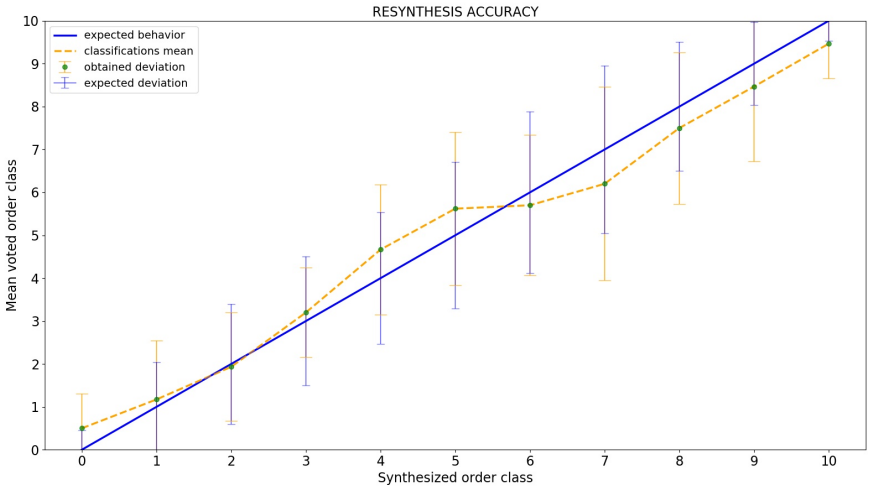

Figure 23: Accuracy of the re-synthesis algorithm

Figure 23 displays the averagely perceived order level of the re-synthesized sounds, compared with the expected behavior. The X axis shows the order class requested to the system, whereas the Y axis exhibits the perceived class. The orange segments represent the actual behavior of the algorithm (identified by the arithmetic mean of the scores collected for each synthesized class), while the blue one indicates the expected performance (the classes actually requested to the algorithm). The vertical segments illustrate the class-wise standard deviations: the blue represent the class-wise tolerance intervals (the mean standard deviations obtained through the dataset-classification survey, and therefore, how much a particular class is intrinsically ambiguous for humans). Conversely, the orange show the perceived ambiguity of the re-synthesized sounds (the standard deviations computed for all classifications of each sound class obtained through the re-synthesis survey). The average perception resides within the tolerance intervals for the classes going from 1 to 9, whereas for class 0 and 10 it is, respectively, immediately superior and inferior.

Nevertheless, since the expected and obtained standard deviations highly overlap for all classes, we consider all results as acceptable. The graph reveals that the algorithm produces textures averagely more chaotic than the requests above class 5, whereas below and for this class the perceived result is slightly more ordered, except for class 2, for which the results averagely coincide with the expectation. The average ambiguity level assessed through the re-synthesis survey is near to the one computed for the dataset-classification one (which was 1.483, that is 29.6%). In fact, the mean standard deviation relative to the entire re-synthesis survey is equal to 1.516 (30.3%, proportioned to the scale range), which oscillates between a minimum of 0.806 (for class 0 and 10) and 2.25 (for class 7).

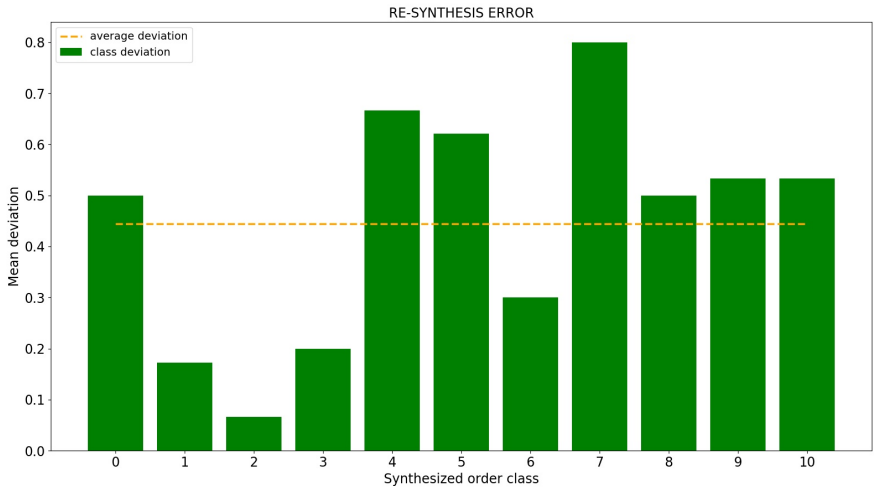

Figure 24: Average error of the re-synthesis algorithm

Figure 24 represents the mean error introduced by the re-synthesis algorithm for each order class. The X axis shows, again, the requested class, whereas the Y axis displays the mean deviation between the requested class and the collected classifications. This value is computed applying:

$$\frac{\sum_{n=0}^{N-1} |X_n - X_r|}{N}$$

where X_n are the individual classifications collected for each requested order class, X_r is the order class requested to the algorithm and N is the amount of classifications collected for the class. The green bars indicate the class-wise mean deviations, while the orange horizontal line shows the overall one (among all classes). The mean deviation, which indicates the mean error of the algorithm, is clearly class dependent, presenting a minimum of 0.067 for class 2 and a maximum of 0.8 for class 7. The overall average error is equal to 0.445, which, proportioned to the possible classification range, indicates an average error percentage of 4.45%.

Nevertheless, it should be considered that the ambiguity level is relatively high for certain classes (especially for class 7), therefore a single sound generated by the algorithm could not match the perception of a single person about the requested class. However, being the average error introduced by the re-synthesis acceptable and the class-wise standard deviation of the re-synthesis and dataset-classification surveys very similar, the re-synthesis algorithm seems to be able to accurately replicate the inquired feature, coherently reproducing its average perception and ambiguity for each class, despite a contained error rate.

At the end of the third stage, we obtained a system capable of producing audible textures matching a desired order class. The re-synthesis model is based on a granular synthesizer connected with a simple Markov process that computes its parameters. Overall, the model is capable of coherently reproducing the chaos/order sound archetype. In particular, the perceived accuracy of the generated sounds has shown to be class-dependent, presenting an average percentage of 95.5% and an average ambiguity level of 30.3% (which reflects the intrinsic perception ambiguity of the inquired feature).

4.4 The user interface

At the end, the classification and re-synthesis algorithms have been connected in a single framework with a simple user interface coded in Max Msp, which is portrayed in Figure 25.

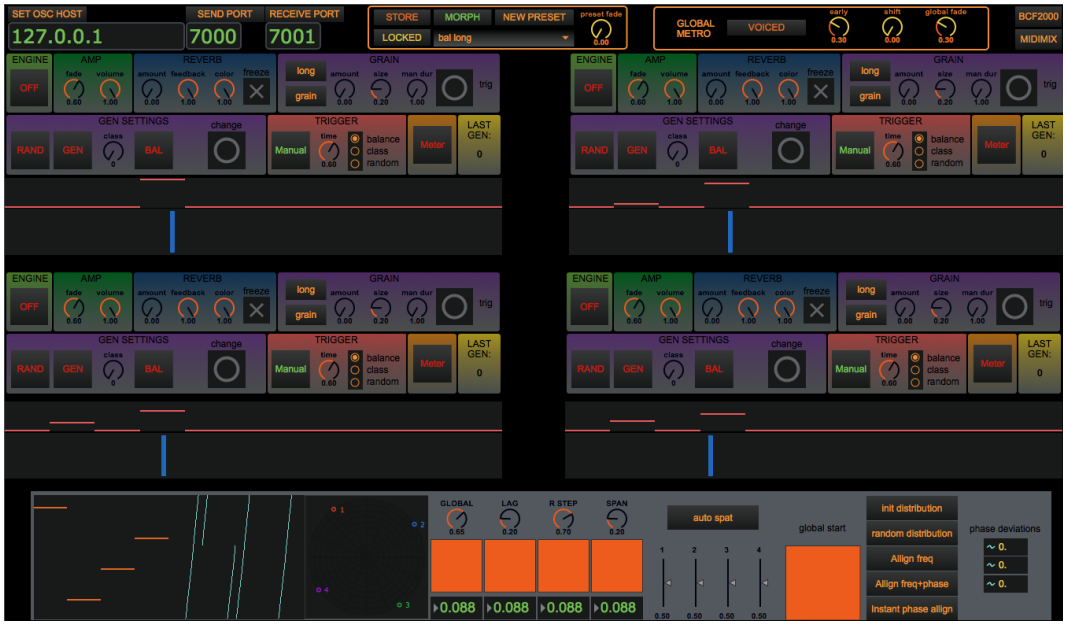

Figure 25: The user interface

The interface, has been arranged for a quadraphonic performance, in order to be able to manage a multichannel and immersive environment. The 4 central blocks are identical and each contains one analysis/re-synthesis channel. Each one provides the possibility of continuously classifying a signal coming from a distinct input of the sound card and send audio information to an individual output channel. By connecting a microphone to a channel's input and a speaker to its output it is possible, for example, to classify the recorded self output of the system diffused by a speaker, mixed with sounds emitted by people present in the same room. It implements, as well, commands for producing on demand textures with a desired amount of order. Moreover, a sliding bar visually indicates the instantaneous order level perceived by each channel. The generation of new sounds can also be

automated through cyclic updates, to obtain constantly changing and interpenetrating textures. Furthermore, it is possible to make the sounds to be generated directly dependent on the CNN predictions, for example producing textures that present *opposite* order levels[56]. When are captured the same sounds diffused by the system, this procedure creates an environment that constantly "listens to itself" and tries to balance the amount of chaos and order in the room, reacting also to noises emitted by the listeners.

Two supplementary audio processing algorithms have been added to each channel: a plate-like reverb and a random amplitude amplifier. The reverb consists of a Max Msp implementation of the famous Miller Puckette's rev3~ abstraction, which has been coded by *Andrea Vigani*. Instead, the random amplifier applies randomly-spaced percussive amplitude envelopes to an incoming signal. We empirically noticed that applying reverberation to a texture always makes the CNN predictions more oriented towards order. Conversely, imposing random amplitude envelopes shifts the predictions towards chaos. Accordingly, through these two simple controls, it is possible to *bias* the CNN, moving the predictions towards a certain direction. Figure 26 shows a detailed description of the main controls available for each channel, subdivided by macro-sections.

56 Symmetrical in relation to the scales's center (5), which coincides with the state of balance between chaos and order.

SECTION	PARAM.	DESCRIPTION
ENGINE	on/off	Switch on or off the synthesis engine
AMP	fade	Cross-fade time between consecutively generated textures
	volume	Overall volume of the channel
REVERB	amount	Balance between reverb and direct signal
	feedback	Reverb duration
	color	Reverb brightness
	freeze	Activate/deactivate infinite sustain behavior
GRAIN	type	Toggle between long/short envelopes
	auto switch	Toggle between automatic/manual envelope triggering
	trig	Manual envelope trigger
	amount	Balance between amp-modulated and direct signal
	size	Fine tune of the automatic envelopes length
	man dur	Fine tune of the manual envelopes length
GEN SETTINGS	rand	Instantly request of a random texture
	gen	Instantly request of a texture matching a specific class
	class	Class to be generated through the GEN button
	bal	Instantly request of a texture opposed to the current prediction
	change	Visualize when a texture change occurs
TRIGGER	auto switch	Toggle between automatic/manual new texture request
	time	Time between automatic new textures requests
	type	Typology of new requested texture (random, specific class or opposite to the current prediction

Figure 26: Controllable parameters of each analysis/re-synthesis channel

In addition to these capabilities, several ambisonics-based[57] spatialization tools have been added to extend the expressive possibilities of the whole system (positioned at the bottom of the interface). The implemented functions permit to perform automatic and manual circular movements for every individual channel, as well as collective random displacements. Moreover, it is possible to save snapshots of the overall system configuration and gently morph between the stored macro-settings of the environment. Finally, every parameter of the interface is predisposed to be controlled through MIDI CC, as well as NRPN messages. This adds the possibility of manipulating and "playing" the environment trough a physical interface, providing the feeling of a real musical instrument. The functioning of the overall system is summarized in the block diagram

57 http://www.ambisonic.net

represented in Figure 27. For simplicity, the scheme refers to one single analysis/re-synthesis channel.

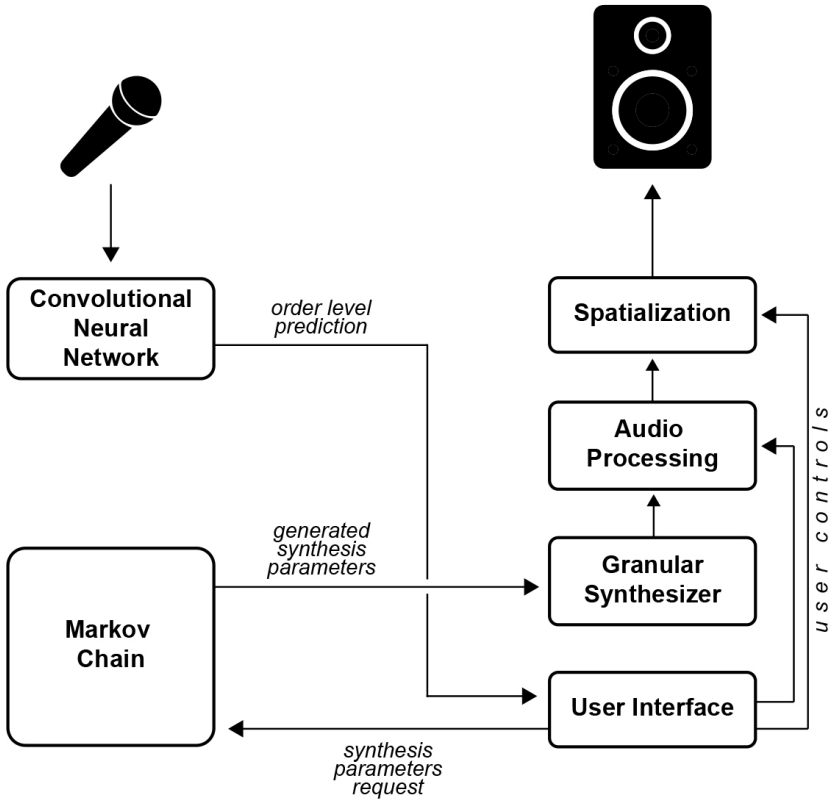

Figure 27: Block diagram of one analysis/re-synthesis channel

As can be inferred from the graph, the analysis and re-synthesis algorithms are independent each other, although they can be connected through user controls. Furthermore, the user can not directly modify the parameters of the granular synthesizer, which can be set only by the Markov chain, responding to specific requests coming from the user interface. Nevertheless, the musician has the possibility of altering the generated textures through the processing algorithms and modifying the spatial mapping of the produced sounds.

At the end of the *method* phase of this research we obtained a working and usable framework to perform archetypical sound analysis and re-synthesis based on the perceptual chaos/order feature. Through the user interface is possible to have direct access to its most important functions in pseudo-real-time. Therefore, the system can be dynamically manipulated with an instantaneous response, providing the feeling of a musical instrument. Furthermore, the achieved algorithm is light-enough to be utilized in a common laptop computer. The overall good analysis and re-synthesis accuracy ensure to have a control on the inquired feature coherent with its average human perception. Accordingly, we consider the primary targets of this research as successfully accomplished.

5 APPLICATION

In order to investigate and demonstrate the expressive potentialities of the developed techniques, we spent approximately one month experimenting its sonic and interaction-related possibilities. At the end, we encoded a performance that goes through the most interesting dimensions achievable with the system, triggering several aesthetic and artistic reflections.

5.1 The performance's structure

The performance is structured around a quadraphonic environment, which is divided in 4 sub-systems, each composed by one microphone and one speaker connected to one individual archetypical analysis/re-synthesis channel. The sub-systems are placed at the corners of an auditory space and, from an operational point of view, are completely independent one another. Nevertheless, they are acoustically interconnected, being physically positioned in the same ambience. Each one is capable of continuously predicting the perceptual sound order level present in a restricted area of the environment, which coincides with the sensible field of the relative microphone. The speakers diffuse continuously mutating textures, which order level is directly influenced by the predictions. The order level captured by one specific microphone can affect only the sounds emitted by its relative speaker (connected to the same channel), modifying the re-synthesis parameters to generate timbres opposed to the predictions (symmetrical in relation to the scales's center). In this way, every sub-system attempts to balance chaos and order in one restricted area of the

ambience, relying on what is captured by its microphone. Every decision taken by a sub-system is aimed at balancing itself, nevertheless it actually affects also the other systems. This happens because they are all acoustically interconnected, and therefore part of an *ecosystem*. The result is an immersive soundscape that constantly chases the balance between chaos and order, sometimes converging into a static point, sometimes diverging into an indefinite state and sometimes oscillating between the two conditions.

The sonic variety of this self-morphing scenario is provided by two primary factors. First, the semi-aleatory process of the re-synthesis guarantee that constantly changing textures are produced. Second, the whole system is affected by an *error rate* (around 3.3% for the analysis and 4.5% for the re-synthesis), which, moreover, is not equal for all order classes. The error manifests itself as inaccuracy in the predictions and, consequently, in the generated textures. The system's task, which is to find a perfect balance between chaos and order in the soundscape, is obviously hampered by this error rate, whereas, from an aesthetic point of view, the environment can benefit from it, generating more variegate and interesting morphings. Therefore, in this context, the intrinsic inaccuracy of the system becomes a characterizing facet of the opus. This ecosystem is thus a vivid *organism,* based on an artificial intelligence algorithm that auto-regenerates its output through an auto-critic feedback that follows rules that reflect the grouping processes proper of human sensorial perception.

The evolution of the piece is based on an interaction of this organism with a human performer and the audience. The first has control over the re-synthesis behavior, whereas the latter can interfere with the system's analysis mechanism. In particular, the performer has the possibility of separately altering each sub-system through the following commands:

- Frequency of the textures' changeover. This controls the regeneration speed, ranging from a very slow and almost static behavior to very quick and syncopate permutations;
- Request of a particular sound class. This permits to momentary disconnect the analysis and re-synthesis algorithms (bypassing the self-balancing process) to demand a texture that presents a desired order level;
- Textures processing. This allows to add reverberation and random amplitude envelopes to the output sounds. As described in the previous section, these procedures can alter the order level of a texture, shifting towards a desired direction and consequently imposing a bias in the predictions;
- Virtual dislocation of speakers. Through the spatialization tools it is possible to virtually move the diffusion of a speaker within the environment. This permits to broaden and morph the sub-systems' operational area, by outdistancing their relative emission and analysis zones.

On the contrary, the audience can emit sounds and occupy the microphones' sensible area, consequently biasing the predictions. Therefore, the piece's execution involves a *dynamic interactivity,* as defined by *Candy and Edmonds* [55]. In fact, both the listeners and the performer have an *active* role in the composition, being able of "influencing the changes in the art object". In particular, the audience can dynamically alter the acoustic balance of the environment, consequently causing sonic mutations in the organism's behavior, whereas the performer, which could be considered as a *supervisor,* can modify the manner in which the organism reacts to the audience and to its self output. Accordingly, in the opus coexist three distinct *agents:*

1. The virtual art object, which is the self-balancing organism;

2. The audience, which can communicate with the latter, visually and sonically adding a "real" component to the virtual art object;
3. The supervisor, which can guide the performance, intermediating between the other agents.

The possibility of "deactivating" these agents, provides different fruition and participation modalities of the opus:

- *Passive and static.* It provides for the deactivation of the agents 2 and 3, eliminating the possibility of any human interaction with the organism. This implicates that no supervisor modifies the environment's behavior and the audience is asked to not emit any sound and not occupy critic analysis areas of the ambience. This leads to a static environment, in which the only possible interaction is the self-influence among the four sub-systems;
- *Passive guided.* This modality provides for the deactivation of the only agent 3. The self interaction of the system is modified as will by the only supervisor. This behaves as a "classic" acousmatic performance, in which the audience listens to a *played* electronic environment;
- *Active guided.* In this modality all agents are active, leading to the involvement of the both audience and supervisor with the art object.

The opportunity of mixing different modalities in the performance provides a further dimension of interaction. The control of this dimension is entrusted to the supervisor, which can "direct" the audience through trivial codified gestures, explained before the execution. Thus, the performance takes the form of an interactive improvisation, in which the decision-making power follows an ascendent hierarchy, from the audience to the artificial intelligence. In fact, the audience's behavior is dependent on the supervisor's decisions and both are submitted to the artificial intelligence's choices. The result is a sort of dialog between humans and the artificial intelligence, in which the AI has always the final word. This means that the

only AI can establish which textures to synthesize, whereas people can just interfere with its decision flow by integrating within its auto-critic feedback. Therefore, the self-reasoning organism can be persuaded by humans to behave in a desired manner, under the guidance of a supervisor that acts as conductor.

5.2 Considerations on the performance

The opus proposes, in the form of a sensorial experience (audible sound), the product of the mental elaboration of sensorial experiences, generating a sort of *feeling the feeling*. As mentioned at the beginning of this document, humans are able to "think sound archetypes", imagining audio fluxes that match desired characteristics. The latter consist of a *biological virtualization* of timbral structures, recognized in experienced sounds. Obviously, it is not possible to directly transform these imaginary audio representations into a collective sensorial experience, since they can not be neither captured, nor recorded, standing at the actual technological development. Nevertheless, they can be externalized through a process of *imitation,* by mimicking the salient characters of an archetypical idea trough musical instruments or, generally, producing isomorphically correlated sounds. A computational simulation of this virtualization process produces numerical results that can be easily transformed into real and audible sound information trough a similar imitation process. In our context, this coincides with the materialization, through the granular synthesis, of the concepts learned by the AI. Thus, the "thought" of an artificial intelligence can be recorded, analyzed and heard. In fact, the textures that compose the opus reflect the manner our AI "thinks" the concept of chaos and order in sound, which is structured around the elaboration of the given experience. The

formal content of the opus is therefore a sensorial manifestation of an artificial thinking that replicates the human virtualization of sensorial experiences.

Even though this computational representation is a mere copycat of human intellection/imagination, it presents several common tracts with the biological sound conception. Above all, it implicates a *non-exact formal determination,* which is obviously intrinsic on the very concept of archetype. In fact, the system does not provide the possibility of forging a timbre following an exactly determined and reproducible process, as opposed to other sound synthesis techniques such as the "classic" frequency modulation or additive. The algorithm permits only to control the textures' perceptual character in retrospect, generating *instances* of a perceptual class, without any control on the most intimate details. This makes the system *non-exactly determinable,* reflecting the (empirically evident) fuzziness of mental re-enactment of sensorial experience. This analogy unifies the artificial intelligence and its human counterpart by identifying a common means, which is the conception of the archetypical idea of chaos and order. The fuzziness of this idea expresses itself in different manners for different intelligent entities, indiscriminately that they are human or artificial. Thus, it is precisely on the reciprocal discovering of this uncertainty character that the interaction of the opus is based. In fact, the AI has been specifically codified to understand and elaborate a human concept, however it introduces an error, which could be considered as the difference that elapses between the average human perception and the AI's specific conceiving of this idea. Therefore, it represents a trait of his "personality". Humans have to investigate and comprehend this error, and then the personality of the AI, in order to permit a proper communication. Hence, during the performance takes place a process of *empathic projection* of the human with the machine and of the machine with the human.

This piece arises in strong connection with the concepts of *sound sculpture* and *auditory ecosystem* elaborated by *Agostino Di Scipio*. The central aspect of his ecosystem-based installations is the identification of the sound information as the *object* of the communication, and not the only *medium,* as provided for the canonical musical composition. The principal manifestation of this concept in the work of the italian composer is the adoption of *feedback networks* to construct a solid and vivid dependance among sound, audience and room's acoustics. This is based on the exploiting of the phenomenon of acoustic *feedback* that naturally occurs by diffusing in the same ambience the sound captured by a microphone. A key aspect of this conception is to impose *algorithmic control* on the feedbacks. This permits to confer a structure to the ecosystem, establishing a guided dialog in which the subjects are the users and the system, whereas the object is the sound. In our opus is present the same feedback-oriented structure and its control coincides with the auto-critic process performed by the AI. Nevertheless, while the subjects of the communication are still the humans (audience and supervisor) and the system, the object is an archetypical auditory construct that is conceived by both subjects. The specific theme of this dialog is the idea of chaos and order, which is intrinsically related to the concepts of unpredictability and interpretability, according to the theory of information. Therefore, the opus arises in relation two intelligent entities of profoundly different nature, establishing a debate regarding their own capability of comprehension.

5.3 Considerations on possible impact and applications

The use of artificial intelligence for artistic purposes is certainly not an innovative development. The relatively dated works of *T. Dartnall* [56] and

M. Boden [57] can give an exhausting overview, as well as a solid theoretical background of AI's creative applications. As regards musical AI implementations, the first appeared in the 1960s, which, due to the technological development of that time, was restricted to the major computing research centers. Two "classic" examples should be mentioned: *Barucha et al.* [58], which in 1989 developed an extraordinary neural-network-based system to model musical schemas and *D. Cope* [59], which in 1980 proposed an AI-based algorithm capable of arranging musical compositions emulating the style of several composers. The advent of increasingly performing and inexpensive computing technology, coupled with the release of simplified software tools for ANN-based processing, made the research on artificial intelligence accessible to the most. Consequently, the possibility of adopting AI-based technologies extended also to artists and, in general, to non specialists. In particular, after the publication of the *Tensorflow* [50] library in 2015, AI-based art and music demonstrated an increasingly rapid diffusion. Nevertheless, notwithstanding the copiousness of studies towards this direction, to our best knowledge, the greater attention has been conferred on modeling the aspect of "musical organization", intended as tonal and temporal dependencies of sounds within musical structures, despite an inner circle of researches such as *Wavenet* [33] and NSynth [34].

With this work, we tried to investigate an approach that, from a technical point of view, does not add any significative innovation to previous methodologies. Although it demonstrates several powerful and interesting implications of AI-based modeling of the *timbre dimension* of sound. Despite the substantial restrictions of the actual environment, the developed performance reveals an interesting expressive approach to music production, improvisation and interaction design. In particular, it allows the artist to create sounds manipulating perceptual and abstract timbral characters. Moreover, it permits to classify any audio signal according to

the same criterion and adopt the predictions to control any parameter of a synthesis or processing algorithm. These processes replicate, respectively, the human imagination and intellection abilities (related to the timbre information of sound), providing novel modalities of artistic experimentation on the concept of human-machine interaction.

Beyond the specific restrictions of the environment we obtained by now, an archetypical sound analysis/re-synthesis framework could be employed for a multitude of different applications. In the first instance, it could lead to the development of new products for industries that produce synthesizers and sound engines. It is reported that the trend recently followed by the most influent companies is to focus on usage immediacy and simplification of the interaction between user and product. The applications derived from our analysis/synthesis framework could be perfectly in line with these tendencies, by offering an intuitive, immediate and powerful approach to sound creation and manipulation. This would have an important impact also in the field of sound design for films and video games. Furthermore, the developed techniques could be adopted for educational purposes, to generate software that stimulate the sound imagination of individuals in the growth phase. Thanks to the inner simplicity of the proposed synthesis method, the child would have an instrument through which imagine and create sounds with a *"block"* conception. By adopting sound archetypes as *"bricks"*, he could construct sounds in an analogous manner as building *Lego* constructions. This would be a valuable stimulus for the sound-related and musical education. In addition, the analysis and classification algorithms developed for this framework could be used for applications aimed to support individuals with communicative disabilities. In particular, it could be employed to automatically catalog and interpret messages generated in the form of non semantically organized sounds. For instance, permitting the interpretation of sounds emitted thorough the vocal tract that do not correspond to real words, or noises to which the subject attributes a

precise meaning. Such a system could help the remote monitoring of people with disabilities.

6 FUTURE WORK

The major weakness of the environment implemented so far can be considered the *non-generalizability*. In fact, only one sound archetype has been modeled and only texture-type sounds have been considered by now. Furthermore, different analysis and re-synthesis architectures could lead to better accuracy and reliability. To improve the automatic signal classification section we intend to investigate different CNN architectures, as well as a RNN-based implementation. The latter, in particular, would be an interesting development, since it would eliminate the restriction of analyzing equally-sized sounds, certainly improving the generalizability of the model. As regards the re-synthesis algorithm, two different approaches will be considered. The first consists of generating through ANNs the synthesis parameters needed by a *Sinusoidal plus Stochastic Model* [28] and performing the actual sound synthesis through the latter. Conversely, the second involves the calculation of the output waveform sample-by-sample with a fully probabilistic model, as observed in the *Wavenet* implementation [33]. These models would certainly extend the re-synthesis possibilities to non-texture-type sounds.

In order to obtain a flexible and generalizable environment, it is fundamental to enlarge the palette of modeled features. The goal is to model a significant amount of low-ambiguity-level characteristics (that can be univocally conceived by different individuals), by constructing other datasets and proposing surveys to classify them. In addition to this, we intend to implement a strategy that permits to model archetypes basing on very restricted datasets. This implies, in particular, to improve our augmentation algorithm and make it reliable for any kind of archetype (instead of the only sound chaos/order*)*. This would provide an artist the possibility to construct his own models with little effort and then to

represent his subjective archetypes, reflecting his personal manner to “think sounds”.

Another valuable achievement would be the implementation of archetype-level processing tools. This would permit to build sounds mixing, morphing and sequencing different archetypes. For this purpose, we will experiment the idea of partitioning the sound object in four fundamental components: *multiband envelope*, *attack texture*, *evolution texture, decay texture,* and separately modeling each component. The first would serve as macro-description of the timbre trend. Conversely, the last three would provide a portrayal of the timbre structure, distinct in three significant temporal moments. The subdivision of a sound archetype model in four separate representations would certainly enlarge the expressive possibilities of the framework. For example, it would allow to create a sound with *“boomy”* attack, *“brassy”* evolution, *“soft”* decay and *“crashed-glass-like”* macro-trend, simply by selecting and meshing the correspondent features from different archetype models.

7 CONCLUSIONS

In this research we illustrated and motivated a sound analysis and re-synthesis technique, which we have called “archetypical”. This approach aims to algorithmically reproduce the human ability of classifying sound timbres according to their perceptual characteristics and of mentally re-evoking sounds matching desired features. In this instance we focused on one single sound archetype: the human-perceived *chaos/order* level in audio information.

In order to obtain a representation of the human perception of this feature, we initially created a dataset containing 1000 different sound textures, produced through a granular synthesis algorithm. We proposed then a formal survey to 80 distinct persons to obtain 4 independent judgements about the perceived order level of each data-point. The inquired feature has been proved to present an ambiguity level of 29.6%, relying on the tester’s classifications and the adopted descriptor (mean standard deviation). Therefore, it can be considered a feature that different individuals perceive in a similar, although not identical, manner. In particular, the survey revealed that humans are more concordant in conceiving extremely chaotic or ordered textures, whereas a higher ambiguity is present for the transitional classes.

To achieve an algorithm capable of automatically identifying and quantifying this perceptual feature in audio signals, we implemented a Convolutional Neural Network model. The latter has been trained with the collected dataset processed through augmentation and segmentation algorithms, providing a test accuracy of 96.7% for our dataset, assessed evaluating the model on data unobserved neither for parameters’ (training), nor for hyperparameters’ (design) adjustment. When tested with new sounds, captured by microphones, the model shows an accurate

performance, demonstrating a satisfactory ability of generalizing the learned concepts.

In order to reproduce the human ability of mentally re-evoking sound timbres matching a desired chaos/order level, we developed a re-synthesis model, based on a granular synthesizer controlled by a Markov chain. The accuracy of the re-synthesized timbres has been assessed through another survey, computing the average deviation between the requested order classes and the human judgements. The perceived accuracy of the generated sounds has shown to be class-dependent, presenting an average percentage of 95.5% and an average ambiguity level of 30.3% (which reflects the intrinsic perception ambiguity of the inquired feature). The model can therefore be considered capable of coherently reproducing the chaos/order sound archetype, relying on the adopted descriptor (average deviation) and on the testers' judgements.

The analysis and re-synthesis algorithms have been connected in a single framework with a simple user interface, which provides access to their most important functions in pseudo-real-time. It brings control over the chaos/order level of a sound to be generated, automation modalities, audio processing and spatialization. Furthermore, it provides a fluent pseudo-real-time computation and visualization of the perceptual order level present in signals captured by the audio interface. The system can be dynamically manipulated with an instantaneous response, providing the feeling of a musical instrument. Furthermore, the achieved algorithm is light-enough to be utilized in a common laptop computer, which was one of the primary objectives of this research.

Finally, we demonstrated a potential use of this technique, encoding an interactive performance entirely based on the obtained framework. The achieved conceptual and technical complexity suggest powerful implications, especially in an artistic context. Notwithstanding the restriction to one single sound archetype, this technique reveals an

appealing approach to music production, sound design, improvisation and interaction design. In particular, it allows the artist to create sounds manipulating perceptual and abstract timbral characters. Moreover, it permits to classify any audio signal according to the same criterion and to adopt the predictions to control any parameter of a synthesis or processing algorithm, providing novel modalities of artistic experimentation on the concept of human-machine interaction.

Considering the obtained accuracy and flexibility, we retain that the primary objectives of this research have been successfully accomplished. Therefore we are strongly encouraged to advance this project, with the prospect of achieving a more generalizable framework, capable of accurately modeling any kind of sound archetype.

8 BIBLIOGRAPHY

[1] Schaeffer P., *Traité des Objets Musicaux,* Editions du Seuil, 1966

[2] Frey A., Hautbois X., Bootz P., Tijus C., *An Experimental Validation of Temporal Semiotic Units and Parameterized Time Motifs,* Musicae Scientie, Vol. 18, pp. 98-123, 2014

[3] Lakoff G., Johnson M., *Metaphors We Live By,* University of Chicago Press, 1989

[4] McAdams S., Bigand E., *Thinking in Sound: The Cognitive Psychology of Human Audition,* pp. 146-198, Oxford University Press, 1993

[5] Vicario G. B., *Prolegomena to the Perceptual Study of Sounds.* Sounding Object, Edizioni di Mondo Estremo, Chpt. 2, 2003

[6] Kao M., Yang C., *Tempo and Beat Tracking for Audio Signals with Music Genre Classification,* Int. J. Intelligent Information and Database Systems, Vol. 3, 2009

[7] Han S., Humphreys G. W., Chen L., *Uniform Connectedness and Classical Gestalt Principles of Perceptual Grouping,* Perception & Psychophysics, Vol. 61, pp. 661-674, 1999

[8] Roelfsema P. R., *Cortical Algorithms for Perceptual Grouping,* The Annual Review of Neuroscience 29:203-27, 2006

[9] Bogdanov D., Wack N., Gomez E., Gulati S., Herrera P., Mayor O., Roma G., Salamon J., Zapata, J., Serra X., *ESSENTIA: An Audio Analysis Library for Music Information Retrieval,* International Society for Music Information Retrieval Conference, pp. 493-498, 2003

[10] Maind S. B., Wankar. P., *Research Paper on Basic of Artificial Neural Network,* International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 2, Iss. 1, 2014

[11] Rocchesso D., Bresin R., Fernström M., *Sounding Objects,* IEEE Computer Society Press Los Alamitos, 2003

[12] Goodfellow I., Bengio Y., Courville A., *Deep Learning,* MIT Press, 2016

[13] Kriesel D., *A Brief Introduction to Neural Networks,* http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf, 2005

[14] Covington P., Adams J., Sargin E., *Deep Neural Networks for YouTube Recommendations,* Proceedings of the 10th ACM Conference on Recommender Systems, 2016

[15] Gounaropoulos A., Johnson. C. G., *A Neural Network Approach for Synthesising Timbres From Adjectives,* Proceedings of the 4th Sound and Music Computing Conference, 2007

[16] Rabiner L. R., *On the Use of Autocorrelation Analysis for Pitch Detection,* IEEE Trans. Acoust., Speech, Signal Processing, Vol. 25, No. 1, 1977

[17] Van den Oord A., Dieleman, S. Schrauwen B., *Deep Content-Based Music Recommendation,* Advances in Neural Information Processing Systems 26 (NIPS), 2013

[18] Balakrishnan A., Dixit K., *Deep Playlist: Using Recurrent Neural Networks to Predict Song Similarity,* Google Scholar, Conference Proceedings, 2016

[19] Antipov G., Berrani S. A., Ruchaud N., Dugelay. J. L., *Learned vs. Handcrafted Features for Pedestrian Gender Recognition,* Proceedings of the 23rd ACM International Conference on Multimedia, pp. 1263-126, 2015

[20] Choi K., Fazekas, G. Sandler M., *Towards Playlist Generation Algorithms Using RNNs Trained on Within-Track Transitions,* arXiv:1606.02096, 2016

[21] Nargesian F, Samulowitz H., Khurana U., Khalil E. B., Turaga D., *Learning Feature Engineering for Classification,* Proceedings of the

26thth International Joint Conference on Artificial Intelligence (IJCAI-17), 2017

[22] Pons J., Lidy T., Serra X., *Experimenting with Musically Motivated Convolutional Neural Networks,* 14th International Workshop on Content-Based Multimedia Indexing (CBMI), IEEE, pp. 1-6, 2016

[23] Stamenovic M., *Identifying Cover Songs Using Deep Neural Networks,* Semanthic Scholar, Conference Proceedings, 2015

[24] Stutz D., *Understanding Convolutional Neural Networks,* Seminar on Current Topics in Computer Vision and Machine Learning, 2014

[25] Jaeger H., *A Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "Echo State Network" Approach,* GMD Report 159, German National Research Center for Information Technology, 2002

[26] Choi K., Fazekas G., Sandler M., Cho K., *Convolutional Recurrent Neural Networks for Music Classification,* Neural and Evolutionary Computing, Cornell University Library, 2016

[27] Zhang Y., Pezeshki M., Brakel P., Zhang S., Laurent C., Bengio Y., Courville A., *Towards End-to-End Speech Recognition with Deep Convolutional Neural Network,* arXiv:1701.02720v1, 2017

[28] Serra X.. *Musical Sound Modeling with Sinusoids plus Noise,* Musical Signal Processing, Swets & Zeitlinger, pp. 91-122, 1997

[29] Kayte S., Mundada M., Gujrathi J., *Hidden Markov Model Based Speech Synthesis: A Review,* International Journal of Computer Applications (0975-8887), Vol. 3, 2015

[30] Xenakis I., *Formalized Music: Thought and Mathematics in Composition,* Special Issue of La Revue Musicale, Nos. 253–254, Editions Richard-Masse, 1963

[31] McDonald K., *Neural Nets for Generating Music,* https://medium.com/artists-and-machine-intelligence/neural-nets-for-generating-music-f46dffac21c0, 2017

[32] Zen H., Senior A., Schuster M., *Statistical Parametric Speech Synthesis Using Seep Neural Networks,* Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7962-7966, 2013

[33] Van den Oord A., Simonyan K., Kalchbrenner N., Dieleman S., Vinyals O., Senior A., Zen H., Graves A., Kavukcuoglu K., *Wavenet: A Generative Model for Raw Audio,* 9th ISCA Speech Synthesis Workshop, pp. 125-125, 2016

[34] Engel J., Resnick C., Roberts A., Dieleman S., Eck D., Simonyan K., Norouzi M., *Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders*. arXiv:1704.01279, 2017

[35] Reed S., Akata Z., Yan X., Logeswaran L., Schiele B., Lee H., *Generative Adversarial Text to Images Synthesis,* International Conference of Machine Learning, 2016

[36] Panzner M., Cimiano P., *Comparing Hidden Markov Models and Long Short Term Memory Neural Networks for Learning Action Representations,* Lecture Notes in Computer Science, Vol. 10122, Springer, Cham, 2016

[37] Schwarz D., *Current Research in Concatenative Sound Synthesis,* Proceedings of the International Computer Music Conference (ICMC), 2005

[38] McDermott J. H., Simoncelli P. E., *Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis,* 71(5):926-940. doi:10.1016/j.neuron.2011.06.032., 2011

[39] Jung C. G., Jung L., Meyer-Grass M., Falzeder E., *Children's Dreams: Notes from the Seminar Given in 1936-1940,* Princeton University Press, 2012

[40] Hensman P., Masko D., *The Impact of Inbalanced Training Data for Convolutional Neural Networks,* Degree project in Computer Science, KTH Royal Institute of Technology, 2015

[41] Roads C., *Microsound,* The MIT Press, 2005

[42] Johns R., *Likert Items and Scales,* Survey Question Bank: Methods Fact Sheet 1, 2010

[43] Shannon C. E., *A Mathematical Theory of Communication,* The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, 1948

[44] Wang J., Perez L., *The Effectiveness of Data Augmentation in Image Classification Using Deep Learning,* arXiv:1712.04621v1, 2017

[45] Wu J., *Introduction to Convolutional Neural Networks,* Stanford cs224n: Deep Learning for Natural Language Processing - Bskog AI, 2017

[46] Hubel D. H., *Transformation of Information in the Cat's Visual System,* Proceedings of the International Union of Physiological Sciences, Vol. 3, 1962

[47] Nair V., Hinton G. E., *Rectified Linear Units Improve Restricted Boltzmann Machines,* Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 807-814, 2010

[48] Salomon J., Bello J. P., *Deep Convolutional Networks and Data Augmentation for Environmental Sound Classification,* IEEE Signal Processing Letters, 2017

[49] Chollet F., *Keras,* https://github.com/fchollet/keras, 2015

[50] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mane D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viegas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,* CoRR abs/1603.04467, 2015

[51] Srivastava N., Hinton G., Krizhevsky A., Sutskever I, Salakhutdinov R . , *Dropout: A Simple Way to Prevent Neural Networks from Overfitting,* Journal of Machine Learning Research 10, pp. 1929-1958, 2014

[52] Kingma D. P., Ba J., *Adam: A Method for Stochastic Optimization,* arXiv:1412.6980, 2014

[53] Tolver A., *An Introduction to Markov Chains,* ISBN 978-87-7078-952-3, 2016

[54] Arcella A., *A Computer Model of the Compositional Process,* Proceedings of the International Symposium Xenakis, La musiqueélectroacoustique / Xenakis, The electroacoustic music, 2012

[55] Candy L., Edmonds E., *Explorations in Art and Technology,* Springer-Verlag London Ltd, 2002

[56] Dartnall. T, *Artificial Intelligence and Creativity,* Kluwer Academic Publications, 1994

[57] Boden M. A., *The Creative Mind: Myths and Mechanisms,* Basic Books, Inc. New York, 1991

[58] Bharucha J. J., Todd P. M., *Modeling the Perception of Tonal Structure with Neural Nets,* Computer Music Journal, Vol. 13, No. 4, 1989

[59] Cope D., *Pattern Matching as an Engine for the Computer Simulation of Musical Style,* Proceedings of the International Computer Music Conference, 1990

8 ACKNOWLEDGEMENTS

First of all I would like to thank Nicola Bernardini and Antonio Rodà for the supervision of this research. Moreover, thanks go to:

- All the persons who participated to the surveys;
- Alberto Novello, for his constant presence and valuable help on about any aspect of this project (from scientific, artistic to writing consultancy);
- Fabio Pasa, for the interesting conversations and food for thought about artistic and semiotic implications of this research;
- Andrea Perizzato, Mauro Masin, Matteo Piazza, Luca Pasa, for the interesting conversations and the scientific consultancy;
- Martino Facchin, for the technical/IT advise.

Finally, special thanks go to my family, which always supported and valorized my work.

For the realization of this project the sequent hardware and software instrumentation has been adopted:

- Desktop computer assembled with 16GB RAM DDR3 1600 MHz, Intel i7 3930k 3.2 GhZ, Nvidia GeForce 1050 Ti, running Ubuntu Linux 16.04 with i3 Tiling Window Manager;
- Laptop computer: Macbook Pro 2011 with Intel i5 2.3 Ghz, 4GB RAM DDR3 1333 MHz. Running OSX 10.9.5;
- 4x Monitors Esio nEar05 classic II;
- Headphones Beyerdynamic DT 770 PRO;
- 4x Behringer C2 and 1x Shure SM57 microphones;
- Soundcard RME Firaface UCX;
- Mixer Roland VM-7100/VM-C7100;

- Python 2.7 extended with the sequent libraries: Keras, TensorFlow, Theano, Numpy, Scipy, Librosa, Essentia, SMSTools, Breadpool, ConfigParser, PyOSC, Queue, Matplotlib;
- Max Msp 7.1 extended with the sequent libraries: ICST Ambisonic Tools, Anvigtools;
- Pro Tools HD 10;
- Libre Office 5.1.

The following sounds, downloaded from the Freesound database, have been processed to build the classification dataset: 220342, 268251, 272166, 274281, 295547, 308419, 322269, 000000, 266651, 266828, 267257, 267342, 268816, 268952, 272061, 278224, 315971, 316750, 316832, 321565, 321597, 321649, 321652, 95253, 120994, 126065, 152682, 165092, 165194, 179882, 190499, 215050, 260234, 260705, 262455, 266329, 266330, 266382, 266681, 266758, 266930, 266969, 267108, 267162, 267175, 267282, 267286, 267296, 267536, 267542, 268537, 268780, 268787, 269348, 269597, 269706, 270001, 271311, 271386, 274232, 274632, 276086, 277161, 277169, 277444, 277511, 277712, 278988, 278990, 280208, 294453, 315648, 315707, 315748, 315902, 316814, 316817, 318203, 318599, 319141, 319181, 319382, 319420, 319490, 319549, 320394, 320438, 321063, 321241, 321504, 321523, 321895, 321989, 322064, 322145, 329655, 376452, 384391, 401721, 402521. We apologize to not cite directly the usernames of the sounds' creators, although it is possible to trace back to their profiles simply inserting the above listed IDs in the search engine of www.freesound.org.