AUTOMATIC RECOGNITION OF TEXTURE IN RENAISSANCE MUSIC

Emilia Parada-Cabaleiro[1,2] **Maximilian Schmitt**[3] **Anton Batliner**[3]
Björn Schuller[3,4] **Markus Schedl**[1,2]
[1]Multimedia Mining and Search Group, Institute of Computational Perception, JKU Linz, Austria
[2]Human-centered AI Group, AI Lab, Linz Institute of Technology (LIT), Austria
[3]Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany
[4] GLAM – Group on Language, Audio & Music, Imperial College London, UK
emilia.parada-cabaleiro@jku.at

ABSTRACT

Renaissance music constitutes a resource of immense richness for Western culture, as shown by its central role in digital humanities. Yet, despite the advance of computational musicology in analysing other Western repertoires, the use of computer-based methods to automatically retrieve relevant information from Renaissance music, e. g., identifying word-painting strategies such as *madrigalisms*, is still underdeveloped. To this end, we propose a score-based machine learning approach for the classification of texture in Italian madrigals of the 16^{th} century. Our outcomes indicate that Low Level Descriptors, such as intervals, can successfully convey differences in High Level features, such as texture. Furthermore, our baseline results, particularly the ones from a Convolutional Neural Network, show that machine learning can be successfully used to automatically identify sections in madrigals associated with specific textures from symbolic sources.

1. INTRODUCTION

The 'classical' Italian madrigal is a secular vocal composition from the 16^{th} century, typically for 4 to 6 vocal parts, characterised by a close relationship between music and text [1]. Due to the great historical value of madrigals for the Western cultural heritage, many initiatives aiming to preserve and investigate this repertoire through computational means have been presented, such as *The Marenzio Online Digital Edition* (MODE) [1] and the *Tasso in Music Project* [2], amongst others [3–6]. Nevertheless, in comparison to other relevant genres from Western repertoires, such as Bach's chorales [7–9] or operas [10–12], the application of machine learning (ML) to the understanding of Renaissance music is still rare [13, 14]. Indeed, the investigation of *madrigalisms*, i. e., the word-painting strategy typical of madrigals [1], has not yet been automatised – a subject identified as of great interest [15].

In the 'classical' Italian madrigal, unlike the madrigal of the 17^{th} century, the meaning of the lyrics is often expressed through textural changes. Due to the prominent role of texture in the *madrigalisms* of these specific madrigals, as a first step to approach this topic, we statistically assess which musical features are involved in different textures. Furthermore, we present baseline results for their classification. The experiments were carried out on the SEILS dataset [16], from which a variety of features related to the time, frequency, and time-frequency dimensions were extracted with the `music21` toolkit [17]. The performance of four ML models, i. e., Support Vector Machines (SVM), Multi-layer Perceptrons (MLP), Convolutional Neural Networks (CNN), and Bidirectional Long-Short Term Memory Recurrent Neural Networks (BLSTM-RNN), was evaluated for recognition of three types of texture: antiphonal (ANT), contrapuntal (CON), and homorhythmic (HOM).

The goal of our study is three-fold: (i) Identify the features characteristic of different textures through the extraction and evaluation of symbolic Low Level Descriptors and statistical functionals; (ii) initialise a research path for automatic recognition of word-painting, as a first step focussed on texture, which later should be followed by the evaluation of *madrigalisms*' textual content; (iii) increase the interest within the ML community in applying artificial intelligence (AI) to digital humanities.

The rest of the manuscript is laid out as follows: Section 2 gives an overview of the related work; Section 3 introduces the considered repertoire; Section 4 and Section 5 outline the feature extraction and evaluation; Section 6 and Section 7 describe the experimental set-up and the ML baseline; Section 8 concludes the paper. To promote further improvements in the field, the source code which enables researchers to replicate the statistical assessment and the baseline results are freely released. [2]

2. RELATED WORK

With the advent of digital humanities in general and computational musicology in particular, more and more sym-

[1] `www.marenzio.org/index.xhtml`

[2] `github.com/SEILSdataset/Texture_Recognition`

bolic musical corpora have been presented in the literature. Some of these are the ELVIS database, [3] the *Kernscores* database [18], the MUTOPIA project [4] database, or the *Digital Interactive Mozart Edition* [19]. The potential of preserving music in a codified syntax has prompted well-defined crowdsourcing initiatives aimed to encode and share symbolic music [20]. In parallel to these, projects focusing on the symbolic codification of Renaissance music, such as the *Tasso in Music Project* [2] have been carried out. Furthermore, given the inherent complexity of codifying early music, specific guidelines, aimed to minimise encoding inconsistencies [21] that may lead to bias in the data and therefore distort the ML outcomes [22], have been presented [23].

Nevertheless, when we consider the symbolic corpora containing annotations, these are considerably reduced [24]; therefore, systems such as *Dezrann* [25], designed to collaboratively collect analytic annotations, have been developed. Concerning Western music in general, annotated symbolic corpora have been presented, e. g., to enable the automatic analysis of harmony [24, 26] and musical structure [27, 28]. Although annotated corpora of Renaissance music have also been presented, these are still much more limited [29–31]. Similarly, computational methods aiming to investigate early music have also been developed, such as the online analysis search functionalities of the *Josquin Research Project* [32] and the SIMSSA Project [6], amongst others [14,33,34]. However, to the best of our knowledge, approaches to automatically extract or retrieve specific attributes typical of early music, such as *madrigalisms*, have not yet been presented.

3. DATA DESCRIPTION

The word-painting strategy used in madrigals to musically imitate the meaning of particular words is known as *madrigalism* [1]. In the 'classical' Italian madrigals of the 16^{th} century, *madrigalisms* can involve rhythm or pitch but are in particular defined by changes in polyphonic texture, for example the alternation between imitative and homophonic counterpoint. Specifically, we will consider three types of texture typically associated to *madrigalisms*: antiphonal texture (ANT), i. e., alternating a musical-linguistic pattern between two parts; contrapuntal texture (CON), i. e., staggering a musical-linguistic pattern along the timeline over the different parts; and homorhythmic texture (HOM), i. e., musical-linguistic patterns occur simultaneously in the different parts. For musical examples and further details on each texture, the reader is referred to [29].

The experiments were carried out on the SEILS dataset [16], a corpus containing 30 symbolically codified madrigals from the *Il Lauro Secco* anthology. This collection is particularly suited to evaluate *madrigalisms*, since this word-painting technique is common for its composers, e. g., Luca Marenzio [1]. All the madrigals in the corpus are written for five parts: Canto, Alto, Quinto, Tenor, and Basso, from the higher to the lower. The modern notated transcriptions codified in **kern syntax were considered. Although four kinds of texture are annotated in the corpus – CON, HOM, ANT, and COMB (combined) – the COMB one, which is a combination of the previous ones, was discarded due to its ambiguity. For simplicity, from now on we will refer to the annotated sections as *madrigalisms*.

Group	LLD	Description
Time	BEAT	Note's position (parsed into time-signature units)
	OFFSET	Note's position (parsed into crotchet units)
	RHYTHM	Note's rhythm (as a fraction of crotchet notes)
Frequency	PS	Pitch space representation (e. g., 60.0 stands for C4)
Time-freq.	INTERVAL	Interval between two notes (expressed in semitone units)
	MUS-TEXT	Binary music-text relationship (syllabic and melismatic)

Table 1. Description of the 6 Low Level Descriptors (LLDs) and their corresponding feature groups.

4. FEATURE EXTRACTION

The extracted features can be grouped into three classes related to three dimensions: *Time*, *Frequency*, and *Time-freq.*, i. e., the combination of the first two. This formulation relates to the 'standard' 2-dimensional score representation typical of written Western music, where *Time* is encoded on the *x* axis and *Frequency* (pitch in music theory) on the *y* axis, as shown in piano-rolls [35]. [5] For each dimension, specific Low Level Descriptors (LLDs), i. e., "Unambiguously defined and objectively verifiable concepts" [36], were extracted with the `python` library `music21` [17]. Note that other formulations of LLDs in symbolic music differing from the herein considered have also been presented [37]. Subsequently, statistical functionals were computed from the LLDs (cf. Section 4.2).

4.1 Low Level Descriptors (LLDs)

For each annotated *madrigalism*, six LLDs, chosen from those most representative of each feature group, were extracted over time considering the 'note' as frame unit: Three LLDs relate to *Time* (BEAT, OFFSET, and RHYTHM); one to *Frequency* (PS); two to *Time-freq.* (INTERVAL and MUS-TEXT); cf. Table 1. BEAT indicates the position of each note according to the time-signature. [6] OFFSET gives the position of each note according to crotchets (standard length unit). [7] RHYTHM is indicated as a fraction of crotchets (represented as 1). PS (pitch space) represents absolute pitches according to the chromatic scale. [8] INTER-

[3] `database.elvisproject.ca/`

[4] `www.mutopiaproject.org/`

[5] Although this representation applies to most of the Western musical notation, exceptions should be considered, e. g., contemporary notation. Note that we are not referring to the musical syntax, e. g., Humdrum.

[6] A 3/4, i. e., a triple simple meter, would be parsed into three crotchet units; a 6/8, i. e., a binary compound meter, into two dotted crotchet units.

[7] For coherence with respect to BEAT and to avoid biasing the features by the score length, the OFFSET was computed within bars' boundaries.

[8] As in MIDI, 60 stands for C4; yet, PS contemplates also microtones and values beyond 0-127, although not present in the evaluated repertoire.

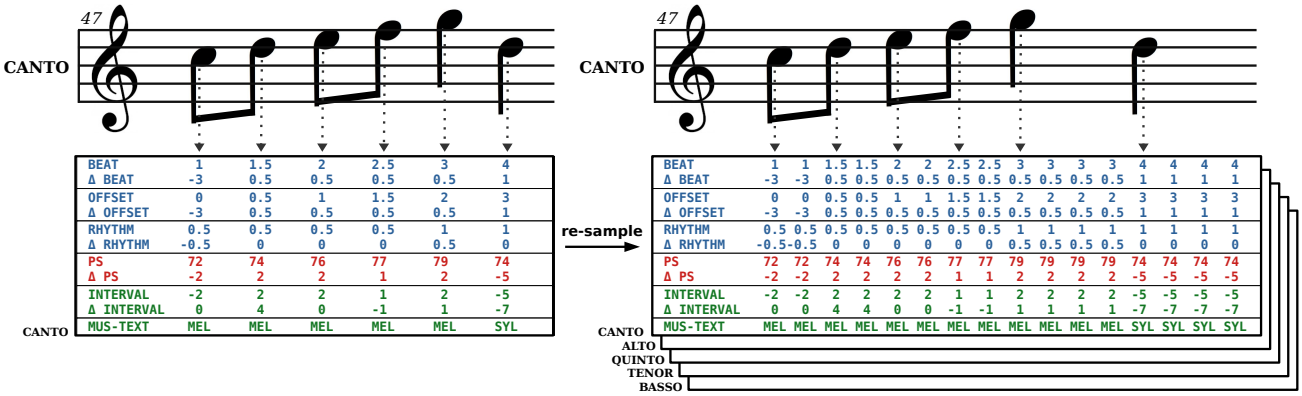

Figure 1. For each part, the Note Level Descriptor (NLD) matrix (on the left) is re-sampled and subsequently 'assembled' into a multi-dimensional NLD (on the right). The example corresponds to bar 47 (Canto part) of Alberti's madrigal.

VAL indicates musical intervals in semitones, where negative values indicate downward intervals, positive upward intervals. MUS-TEXT expresses the (categorical) music-text relationship: either syllabic (one-to-one correspondence between pitches and text syllables) or melismatic (more than one pitch corresponding to each text syllable).

Although in Figure 1, MUS-TEXT is encoded categorically for comprehensibility (SYL as syllabic, MEL as melismatic), in the LLDs, it is encoded as: 0 for no text, i. e., rests; 1 for syllabic; -1 for melismatic. As standard procedure in over-time feature extraction [38], for the 5 continuous LLDs (all except for MUS-TEXT), Delta coefficients (Δ), i. e., the differences between two consecutive values, were computed. From now on, we refer to the 6 LLDs and the 5 Δ as Note Level Descriptors (NLDs). Δ PS and INTERVAL are redundant; yet, they were extracted because they belong to different groups and are thus needed in the statistical analysis. Note that Δ BEAT and Δ OFFSET are only redundant when the subdivision unit of the time-signature is a crotchet note.

The NLDs were extracted (i) for all the parts together, i. e., 1 NLD matrix per *madrigalism*; (ii) for each part individually, i. e., 5 NLD matrices per *madrigalism*. The former were extracted through the `.flat` property of `music21` (which disregards the vertical alignment across parts) with the only purpose of computing the statistical functionals;[9] the latter are assembled together into multi-dimensional NLDs, by this preserving the correspondence between parts over time, which is relevant to musical texture, thus, also to *madrigalisms*. Since each part presents a unique note's configuration, considering the note as frame unit leads to NLD matrices with different lengths across parts. Thus, in order to assemble them, the NLD matrices were re-sampled to the fraction of the shortest note in the corpus, i. e., a semiquaver (cf. Figure 1).

4.2 Statistical Functionals

For the 10 continuous NLDs, 16 functionals were extracted (cf. Table 2). Since for INTERVAL and Δ INTERVAL, the functionals were extracted considering positive and negative values separately, a total of 12 continuous descriptors were used (i. e., 6 LLDs + 6 Δ). This is necessary to obtain a meaningful result, otherwise the upward and downward intervals are mutually cancelled. For the categorical NLD (MUS-TEXT), 3 functionals were extracted: N(umber) of syllabic notes (N_{syl}); N of melismatic notes (N_{mel}); and the ratio between N_{syl} and N_{mel} ($SYL\text{-}MEL_{ratio}$). All in all, 195 functionals were computed: 192 from the continuous NLDs (16 functionals $\times$ 12 continuous descriptors); 3 from the categorical NLD (3 functionals $\times$ 1 NLD).

Category	Description
Extremes	Maximum, minimum, range
Means	Arithmetic, harmonic, geometric
Moments	Standard deviation, variance, kurtosis, skewness, coefficient of variation
Percentiles	Median, 1^{st} quartile, 3^{rd} quartile, interquartile range
Other	Mode

Table 2. Description of the 16 statistical functionals extracted from the continuous NLDs.

5. FEATURE EVALUATION

5.1 Feature Groups Comparison

To evaluate whether the chosen feature groups are suitable to differentiate between the *madrigalism* classes (ANT, CON, HOM), Welch-ANOVA was considered, an alternative to the one-way 'classic' ANOVA (analysis of variance), suitable in this case: The data were normally distributed but the homogeneity assumption was violated [39]. For the pairwise comparisons across classes, the Games-Howell post-hoc test was employed. Since p-values as evaluation criteria have been repeatedly criticised [40], they will be reported but the statistical outcomes will be interpreted in terms of effect size [41]: epsilon squared (ϵ^2) for the Welch-ANOVA and Hedge's g for the post-hoc test. To enable the comparison across the three feature groups, Principal Component Analysis (PCA) was applied as a method for dimensionality reduction to the functionals' vectors of each group. Although PCA implies information loss, this is a plausible method which enabled us to perform an overall assessment. The remaining variances were: 80 % for *Time*; 67 % for *Frequency* and *Time-freq.*

[9] For the functionals the correspondence between parts is irrelevant.

group	F	df1	df2	p	ϵ^2	ANT–CON				ANT–HOM				CON–HOM			
						lwr	upr	p	g	lwr	upr	p	g	lwr	upr	p	g
Time	6.4	2	410	.002	.02	−1.47	1.67	.988	0.02	−3.51	0.04	.057	0.33	−3.05	−0.61	.001	0.40
Frequency	1.5	2	410	.216	.01	−1.08	1.28	.977	0.03	−0.60	1.98	.413	0.19	−0.24	1.42	.220	0.19
Time-freq.	9.9	2	410	.000	.03	−1.27	1.00	.956	0.04	0.68	3.68	.002	0.42	1.08	3.55	.000	0.52

Table 3. Welch-ANOVA and Games-Howell results for the evaluation of the *madrigalisms*: antiphonal (ANT), contrapuntal (CON), homorhythmic (HOM); for each feature group: *Time*, *Frequency*, *Time-freq.* F statistic, degrees of freedom (df1 and df2), p-values, epsilon-squared (ϵ^2), Hedge's g, confidence intervals: lower (lwr) and upper (upr), are given.

functional	H	df1	df2	p	η^2	ANT–CON			ANT–HOM			CON–HOM		
						Z	p	d	Z	p	d	Z	p	d
N_{syl} (*syllabic*)	76.6	2	410	.000	.18	−1.57	.167	0.29	4.76	.000	0.87	8.66	.000	0.88
N_{mel} (*melismatic*)	26.7	2	410	.000	.06	−0.83	.406	0.10	2.85	.006	0.64	5.04	.000	0.59
SYL-MEL_{ratio} (*ratio*)	18.67	2	410	.000	.04	−0.98	.326	0.19	2.17	.045	0.26	4.30	.000	0.41

Table 4. Kruskal-Wallis results and Dunn pairwise comparisons for the evaluation of the *madrigalism* classes: ANT, CON, HOM; for the three MUS-TEXT statistical functionals: N(umber) of syllabic and melismatic notes, and the ratio between both. H statistic, degrees of freedom (df1 and df2), p-values, eta-squared (η^2), Z-score, and Cohen's d, are given.

The statistical analysis shows that the *Time-freq.* features present the most prominent differences across *madrigalism* classes, as indicated by a higher (although small) effect size with respect to the other feature groups ($\epsilon^2 = .03$); cf. ϵ^2 for *Time-freq.* in Table 3. This difference is medium for CON vs HOM ($g = 0.52$), slightly lower for ANT vs HOM ($g = 0.42$), and almost no difference is displayed for ANT vs CON ($g = 0.04$); cf. g for *Time-freq.* in Table 3. The same trend is displayed to a lower extent for *Time*: higher differences are shown for CON vs HOM and ANT vs HOM ($g = 0.40$ and $g = 0.33$, respectively); almost no difference is shown for ANT vs CON ($g = 0.02$); cf. g for *Time* in Table 3. Conversely, all the differences between classes are small for the feature group *Frequency* ($g \leq 0.19$) which indicates that there is no relationship between *madrigalisms*' texture and specific vocal registers. Nevertheless, the role of frequency-related features should be further investigated by considering the meaning and relevance of the words used in each *madrigalism* class.

Overall, the statistical evaluation indicates that HOM and CON are the *madrigalisms* with the highest dissimilarity, while ANT and CON are the most similar ones. This might seem obvious if we think of the *madrigalisms*' texture, i. e., by evaluating them from a High Level perspective: Contrapuntal and homorhythmic textures are dissimilar; contrapuntal and antiphonal textures are similar. Yet, our analysis indicates that the functionals related to the *Time* and *Time-freq.* dimensions capture relevant properties in the definition of the evaluated classes; consequently, their NLDs are also representative of *madrigalisms*' inherent texture. This shows a direct relationship between Low Level and High Level musical descriptors, meaning that measuring the former may enable us to predict the latter.

5.2 Music-Text Relationships

Since the relationships between music and lyrics are crucial in *madrigalisms*, the functionals extracted from the MUS-TEXT NLD (N_{syl}, N_{mel}, and SYL-MEL_{ratio}) are evaluated individually. Note that these are already vectors, thus, PCA was not performed. Since the assumptions for normality and homogeneity were both rejected, the rank-based non-parametric Kruskal-Wallis test was carried out. For the pairwise comparisons across classes, the Dunn post-hoc test with Benjamini-Hochberg (BH) p-value adjustment was applied [42]. Again, the statistical outcomes will be evaluated in terms of effect size: eta-squared (η^2) for Kruskal-Wallis and Cohen's *d* for the Dunn test.

Our analysis shows that the functionals related to the counts of each MUS-TEXT relationship (syllabic and melismatic) are relevant to differentiate between *madrigalism* classes, as shown by the large and moderate effect sizes, respectively: $\eta^2 = .18$ (for N_{syl}); $\eta^2 = .06$ (for N_{mel}); cf. η^2 in Table 4. Differences between classes are less prominent for the ratio, as shown by a lower effect size ($\eta^2 = .04$); cf. η^2 for SYL-MEL_{ratio} in Table 4. Similarly to the outcomes from the overall evaluation (cf. Section 5.1), HOM shows generally noticeable differences with respect to the other two classes. The pairwise comparisons for CON vs HOM and ANT vs HOM indicate big differences for N_{syl} ($d = 0.88$ and $d = 0.87$); moderate for N_{mel} ($d = 0.59$ and $d = 0.64$); smaller (as expected) for SYL-MEL_{ratio} ($d = 0.41$ and $d = 0.26$); cf. Cohen's d for CON-HOM and ANT-HOM, respectively, in Table 4. Again, ANT vs CON yielded small differences for all the functionals ($0.10 \leq d \leq 0.29$); cf. d for ANT-CON in Table 4.

Since the music-text relationships might particularly vary across the different parts, statistical functionals were also extracted from the MUS-TEXT NLD, considering each part individually; [10] then, the same evaluation was carried out. The results of the statistical analysis for the individual parts, although showing smaller effects, display the same overall trend as described for the parts together. For N_{syl} and N_{mel} (in all the parts), HOM vs the other two classes yielded a moderate effect size ($0.41 \leq d \leq 0.71$), for ANT vs CON a small one ($0.03 \leq d \leq 0.34$). Similarly, for SYL-MEL_{ratio} (in all the parts), all the pairwise comparisons yielded $d \leq 0.40$, except for *Canto*, which showed a slightly higher difference for HOM vs the other two classes ($0.45 \leq d \leq 0.56$). This is due to the *Canto*'s prominent use of melismas in CON and ANT, which – contrasting

[10] The functionals were again extracted before the re-sampling, but in this case, processing the 5 NLD matrices of each *madrigalism* separately. Note that due to space constraints these results are not displayed in a table.

Sp.	Fold A			Fold B			Fold C		
	ANT	CON	HOM	ANT	CON	HOM	ANT	CON	HOM
I	19	58	47	24	79	39	21	64	62
II	22	69	33	26	70	60	16	62	55
III	16	71	42	19	73	58	29	57	48
IV	21	72	51	25	60	49	18	69	48

Table 5. Number of *madrigalisms* per class (ANT, CON, HOM), in the four splittings (Sp.), performed according to the 3-fold composer independent partitioning (A, B, C).

with the syllabism typical of HOM – makes the differences in SYL-MEL$_{ratio}$ across classes much more prominent for this part. This suggests that the role of specific features might be more clearly displayed in some parts, indicating that a particular weight should be attributed to them.

6. EXPERIMENTAL SET-UP

To create the baseline for the automatic recognition of *madrigalisms*' texture, four ML models were considered: a Support Vector Machine (SVM) classifier, a Multilayer Perceptron (MLP), a Convolutional Neural Network (CNN), and a Bidirectional Long-Short Term Memory Recurrent Neural Network (BLSTM-RNN). The SVM and the MLP were fed with the statistical functionals (cf. Section 4.2), the CNN and the BLSTM-RNN with the multi-dimensional NLDs (cf. Section 4.1). Both feature representations were z-score normalised according to the mean and variance, estimated from the respective training set. In addition, a FUSION approach considering the NLDs and functionals was investigated (cf. Figure 2).

6.1 Partitions, Experiments, and Evaluation Metrics

The experiments were carried out on a 3-fold composer independent partitioning, i. e., the *madrigalisms* were split into 3 disjunct sub-sets (A, B, C), and no *madrigalisms* by the same composer appeared across sub-sets. Note that each madrigal is by a unique composer. The 30 composers were randomly [11] assigned to the 3 sub-sets (10 for each), which were considered alternately as training, validation, and test sets. To generalise the outcomes, the 3-fold partitioning was performed 4 times (cf. Table 5), and the experiments were carried out for each of the 4 splits individually. Furthermore, the 6 possible permutations between sub-sets were considered per split; thus, a total of 24 experiments was carried out: 6 permutations $\times$ 4 splits.

As the features from the *Frequency* group proved not to be relevant in the statistical analysis, the following experiments were performed: (i) using the whole feature set (*All*), i. e., 195 functionals (for SVM and MLP) and 11 NLDs (for CNN and BLSTM-RNN); (ii) excluding the features from the *Frequency* group (*Selected*), i. e., 163 functionals and 9 NLDs. Finally, given the high similarity between ANT and CON, two classification problems were addressed: (i) 3C(lass), i. e., considering the three types of *madrigalisms*; (ii) 2C, i. e., excluding ANT (the minority class). Thus, the 24 experiments were performed in four set-ups: 3C with *All* features; 3C with the *Selected* ones; 2C with *All* features; 2C with the *Selected* ones; i. e., 96 experiments (4 splits $\times$ 6 permutations $\times$ 4 set-ups) were carried out per model. To enable a fair comparison, the models were optimised individually for each set-up.

[11] A fixed random seed was chosen to guarantee reproducibility.

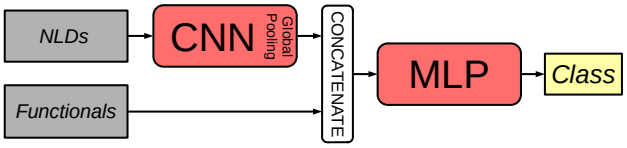

Figure 2. FUSION model: NLDs are fed into a CNN (1 or 2 convolutional blocks and global pooling over time), concatenated with the functionals, and then fed into an MLP.

Since the distribution of *madrigalisms* across composers is unbalanced (cf. Table 5), the samples belonging to the minority classes were up-sampled in training by randomly [11] duplicating *madrigalisms* until matching the size of the majority class, i. e., CON. Unweighted Average Recall (UAR) was considered as appropriate metric to evaluate the models' performance due to the unbalanced classes in the test set; furthermore, the recall for each class will be discussed. To report the overall results, mean (μ) and standard deviation (σ) across the 24 experiments per set-up and model will be indicated for both UAR and recall.

6.2 Model Optimisation

We employed an SVM classifier with linear kernel built on the `python` library `scikit-learn` [43]. For the optimisation, five different complexities (C) from 0.00001 to 0.1 (on a logarithmic scale, with a factor of 10 between steps) were considered. The C which yielded the highest UAR on the validation set was chosen to re-train the SVM considering the samples from the training and validation sets together. The MLP, CNN, and BLSTM-RNN were built on `TensorFlow 2.3` [44] through the API `Keras` [45]. For all of them, Adam optimiser, Softmax activation function in the output layer, a maximum of 200 epochs, and early stopping with a patience of 15 were used.

For the MLP, an architecture of two hidden layers with an optimised number of neurons for each given as 25, 75, or 175 and Sigmoid activation, with a dropout of 20 % after the first hidden layer, was considered. The batch size was optimised choosing the optimum of 10, 25, and 75; the learning rate was fixed as 0.001. All optimisations and the early stopping were done on the validation set. For the CNN and BLSTM-RNN models, the multi-dimensional NLDs (time $\times$ part $\times$ NLD) were first reshaped fusing the part and NLD dimensions. [12] Both front-ends were followed by a fully-connected network where the same architecture and hyperparameters as for the MLP were used.

The CNN front-end consisted of convolutional blocks with a 1D-convolutional layer of 150 filters followed by batch normalisation, ReLU, and a max-pooling layer. For the convolutional layer, the filter length was 3 and the shift 2; for the max-pooling layer, both the filter length and the shift were 2. The number of convolutional blocks was optimised between one and two. The sequential represen-

[12] Using different *heads* for each part was also tried in initial experiments, but the performance was found to be worse.

	All (Time + Freq. + Time-freq.)				*Selected (Time + Time-freq.)*				
3C	SVM	MLP	CNN	BLSTM-RNN	SVM	MLP	CNN	BLSTM-RNN	FUSION
ANT	13.0 ± 8.5	14.3 ± 10.4	8.2 ± 7.5	**24.0** ± 20.8	13.5 ± 7.0	9.4 ± 9.4	9.6 ± 7.1	**19.7** ± 14.8	10.2 ± 8.5
CON	61.5 ± 7.2	57.0 ± 11.8	**74.3** ± 11.3	51.1 ± 16.5	62.3 ± 7.5	65.5 ± 11.9	**72.4** ± 9.6	55.9 ± 13.3	68.1 ± 8.1
HOM	58.4 ± 5.9	59.4 ± 7.9	**70.5** ± 11.8	69.7 ± 16.7	59.7 ± 5.9	59.9 ± 8.8	**70.8** ± 12.0	69.8 ± 9.7	67.1 ± 9.0
UAR	44.3 ± 3.0	43.6 ± 4.0	**51.0** ± 4.6	48.3 ± 5.6	45.2 ± 3.3	44.9 ± 4.0	**50.8** ± 4.4	48.4 ± 5.0	48.5 ± 3.9
2C	SVM	MLP	CNN	BLSTM-RNN	SVM	MLP	CNN	BLSTM-RNN	FUSION
CON	74.4 ± 6.0	72.8 ± 6.1	**80.0** ± 10.3	74.0 ± 11.6	75.9 ± 6.7	75.0 ± 5.9	77.5 ± 10.0	73.0 ± 11.2	**80.3** ± 8.6
HOM	61.4 ± 8.2	62.1 ± 8.2	**71.0** ± 9.6	69.3 ± 11.5	62.7 ± 8.4	61.3 ± 8.1	**74.2** ± 9.2	70.3 ± 8.0	68.1 ± 10.2
UAR	67.9 ± 3.6	67.4 ± 3.8	**75.5** ± 4.5	71.7 ± 6.3	69.3 ± 3.5	68.2 ± 4.6	**75.9** ± 4.4	71.7 ± 5.6	74.2 ± 5.0

Table 6. Baseline results for the 3C(lass) and 2C classification of *madrigalisms* (ANT, CON, HOM) for *All* and *Selected* features. Recall per class and Unweighted Average Recall (UAR) are given (highest values marked in bold) for SVM, MLP, CNN, BLSTM-RNN, and FUSION approaches. Mean and standard deviation ($\mu \pm \sigma$) [%] across experiments are indicated.

tations extracted by the CNN front-end are subject to a global max-pooling (over time). The BLSTM-RNN front-end consisted of BLSTM layers of 150 units, a dropout of 20 %, and Tanh activation. The number of layers was again optimised between one and two. When using two layers, the first layer returned a sequential output, followed by self-attention (`SeqSelfAttention` layer).

7. BASELINE RESULTS

In Table 6, the baseline results are given. Generally, the experiments with *Selected* features present a higher UAR than those with *All* features, which confirms the outcomes of the statistical evaluation (cf. Section 5.1); still, the differences for *Selected* vs *All* for any classifier are small (≤ 1.4 %). The model reaching the highest UAR was the CNN (cf. UAR for CNN in Table 6), generally showing a statistically significant difference with respect to the others. Pairwise comparisons with Tukey post-hoc for CNN vs SVM and CNN vs MLP, in all the set-ups, yielded: $p < .0001$, Cohen's $d > 1.3$; for CNN vs BLSTM-RNN, in 2C: $p < .05$, Cohen's $d > 0.7$; while in 3C, no significant difference was shown: $p > .05$, Cohen's $d < 0.5$.

The higher performance of the CNN, and to some extent BLSTM-RNN, might be due to the use of the NLDs, which unlike the functionals contain the correspondences across parts over time, which is relevant in *madrigalisms*. While the CNN generally performed best with one convolutional block (chosen in 84 out of the 96 experiments), the BLSTM-RNN performed best with two layers (65 out of 96). Yet, the BLSTM-RNN generally shows more unstable results across experiments, as displayed by a higher std. dev. (cf. σ in Table 6), which indicates that a simpler architecture can more reliably model the considered data.

The class recognised worst was, as expected, ANT, showing a recall, for all the models, at chance level (cf. μ for ANT in Table 6). This is due to ANT *madrigalisms* being much fewer than the others and very similar to CON, as shown in the features evaluation. The confusion between both classes particularly reduces the recall of CON in the 3C problem, whose improvement is much more prominent than the one shown for HOM when comparing the 3C and the 2C experiments (cf. the upper with respect to lower half of Table 6 for CON and HOM): Across all the models and feature sets, the average recall difference for CON between 2C and 3C is 12.85 %, while for HOM, it is only 1.76 %.

To evaluate whether a FUSION between the multi-dimensional NLDs and the functionals might yield a better performance, the best performing architecture, i. e., the CNN, was concatenated with the functionals, using the same architecture for the MLP as before (cf. Figure 2). The whole network was trained from scratch to enable the model to learn complementary representations. Experiments were carried out considering the same hyperparameter optimisation as previously described and the same four set-ups: 3C and 2C, for *All* and *Selected* features. The same pairwise combinations between functionals and NLDs were used: *Selected* functionals were concatenated with the output of the CNN trained with *Selected* multi-dimensional NLDs, and correspondingly for *All*. In Table 6, the best results for 3C and 2C problems with the FUSION model, i. e., considering the *Selected* features, are given. While FUSION's recall on CON (2C) increased over the one from the CNN, no consistent improvement is shown, which indicates that multi-dimensional NLDs are a good representation of *madrigalisms*' texture on their own.

8. LIMITATIONS & CONCLUDING REMARKS

Our research outcomes indicate that symbolic features and ML methods are both appropriate to further investigate word-painting strategies in madrigals. They also highlight the potential of applying AI in the study of Renaissance music. Yet, since this study was the first of its kind, at this stage we evaluated the lyrics only in terms of syllabic and melismatic relationship, while the importance of specific words, which might be given by their meaning (both linguistic/metaphorical) within and across madrigals, typically highlighted through specific word-painting strategies, was not yet considered. A deeper evaluation of the lyrics is indeed one of the next priorities in our future work, by this systematically identifying the connections between music and poetry in the Italian madrigal. Furthermore, we will also compare the ML outcomes from the feature-based methods with those achieved through humdrum-based end-to-end approaches already presented in the literature [46].

Our work shows that symbolic Low Level Descriptors are suitable to automatically identify different textures in Italian madrigals. In addition, the presented baseline will hopefully stimulate further research advances in the application of ML to early music, by this promoting a deeper understanding of the Renaissance musical heritage.

9. REFERENCES

[1] W. Apel, *The Harvard dictionary of music*. Cambridge, MA, USA: Harvard University Press, 2003.

[2] E. Ricciardi and C. Sapp, "Editing Italian madrigals in the digital world: The Tasso in Music Project," in *Proc. of Music Encoding Conference*, Virtual event, 2020.

[3] L. Pugin, "Editing Renaissance music: The Aruspix Project," in *Beihefte zur Editio: Internationales Jahrbuch für Editionswissenschaften*. Tübingen: Max Niemeyer, 2009, pp. 94–103.

[4] F. J. Castellanos, J. Calvo-Zaragoza, and J. M. Inesta, "A neural approach for full-page optical music recognition of mensural documents," in *Proc. of the International Society for Music Information Retrieval Conference*. Virtual event: ISMIR, 2020, pp. 23–27.

[5] D. Rizo, J. Calvo-Zaragoza, and J. M. Iñesta, "Muret: A music recognition, encoding, and transcription tool," in *Proc. of the International Conference on Digital Libraries for Musicology*, Paris, France, 2018, pp. 52–56.

[6] G. Vigliensoni, A. Daigle, E. Liu, J. Calvo-Zaragoza, J. Regimbal, M. A. Nguyen, N. Baxter, Z. McLennam, and I. Fujinaga, "From image to encoding: Full optical music recognition of Medieval and Renaissance music," in *Proc. of Music Encoding Conference*, Vienna, Austria, 2019.

[7] Y. Ju, S. Howes, C. McKay, N. Condit-Schultz, J. Calvo-Zaragoza, and I. Fujinaga, "An interactive workflow for generating chord labels for homorhythmic music in symbolic formats," in *Proc. of the International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, 2019, pp. 862–869.

[8] A. Leemhuis, S. Waloschek, and A. Hadjakos, "Bacher than Bach? On musicologically informed AI-based Bach chorale harmonization," in *Proc. of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Würzburg, Germany: Springer, 2019, pp. 462–469.

[9] H. Hild, J. Feulner, and W. Menzel, "Harmonet: A neural net for harmonizing chorales in the style of J. S. Bach," in *Advances in Neural Information Processing Systems*, 1992, pp. 267–274.

[10] M. Krause, F. Zalkow, J. Zalkow, C. Weiß, and M. Muller, "Classifying leitmotifs in recordings of Operas by Richard Wagner," in *Proc. of the International Society for Music Information Retrieval Conference*. Virtual event: ISMIR, 2020, pp. 473–480.

[11] E. Parada-Cabaleiro, M. Schmitt, A. Batliner, S. Hantke, G. Costantini, K. Scherer, and B. Schuller, "Identifying emotions in Opera singing: Implications of adverse acoustic conditions," in *Proc. of the International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 276–382.

[12] C. Brazier and G. Widmer, "Addressing the recitative problem in real-time Opera tracking," *arXiv preprint arXiv:2010.11013*, 2020.

[13] C. Antila and J. Cumming, "The VIS framework: Analyzing counterpoint in large datasets," in *Proc. of the International Society for Music Information Retrieval Conference*. Taipei, Taiwan: ISMIR, 2014, pp. 71–76.

[14] A. Brinkman, D. Shanahan, and C. Sapp, "Musical stylometry, machine learning and attribution studies: A semi-supervised approach to the works of Josquin," in *Proc. of the Biennial International Conference on Music Perception and Cognition*, San Francisco, CA, USA, 2016, pp. 91–97.

[15] C. Sapp, "Suggestions for future corpus-based text painting analyses: A response to Strykowski," *Empirical Musicology Review*, vol. 11, no. 2, pp. 120–123, 2017.

[16] E. Parada-Cabaleiro, A. Batliner, A. E. Baird, and B. Schuller, "The SEILS dataset: Symbolically encoded scores in modern-early notation for computational musicology," in *Proc. of the International Society for Music Information Retrieval Conference*. Suzhou, China: ISMIR, 2017, pp. 575–581.

[17] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. of the International Society for Music Information Retrieval Conference*. Utrecht, Netherlands: ISMIR, 2010, pp. 637–642.

[18] C. Sapp, "Online database of scores in the Humdrum file format," in *Proc. of the International Society for Music Information Retrieval Conference*. London, UK: ISMIR, 2005, pp. 664–665.

[19] I. Cividini, "Zwischen klassischer Musikphilologie und angewandter Informatik: Die Digitale Mozart-Edition (DME) der Stiftung Mozarteum Salzburg," in *Jahrestagung der Gesellschaft für Musikforschung*, Paderborn / Detmold, Germany, 2019.

[20] M. Gotham, P. Jonas, B. Bower, W. Bosworth, D. Rootham, and L. VanHandel, "Scores of scores: An openscore project to encode and share sheet music," in *Proc. of the International Conference on Digital Libraries for Musicology*, Paris, France, 2018, pp. 87–95.

[21] N. Nápoles, G. Vigliensoni, and I. Fujinaga, "Encoding matters," in *Proc. of the International Conference on Digital Libraries for Musicology*, Paris, France, 2018, pp. 69–73.

[22] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[23] J. Cumming, C. McKay, J. Stuchbery, and I. Fujinaga, "Methodologies for creating symbolic corpora of

Western music before 1600," in *Proc. of the International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 491–498.

[24] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula, "Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis," in *Proc. of the International Society for Music Information Retrieval Conference*. Málaga, Spain: ISMIR, 2015, pp. 721–734.

[25] M. Giraud, R. Groult, and E. Leguy, "Dezrann, a web framework to share music analysis," in *Proc. of the International Conference on Technologies for Music Notation and Representation*. Montreal, QC, Canada: TENOR, 2018, pp. 104–110.

[26] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, "The annotated Beethoven corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets," *Frontiers in Digital Humanities*, vol. 5, 2018.

[27] P. Allegraud, L. Bigo, L. Feisthauer, M. Giraud, R. Groult, E. Leguy, and F. Levé, "Learning sonata form structure on Mozart's string quartets," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 82–96, 2019.

[28] M. Giraud, R. Groult, and F. Levé, "Computational analysis of musical form," in *Computational Music Analysis*. Springer, 2016, pp. 113–136.

[29] E. Parada-Cabaleiro, M. Schmitt, A. Batliner, and B. W. Schuller, "Musical-linguistic annotations of Il Lauro Secco," in *Proc. of the International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 461–467.

[30] R. de Valk, R. Ahmed, and T. Crawford, "JosquIntab: A dataset for content-based computational analysis of music in lute tablature," in *Proc. of the International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, 2019, pp. 431–438.

[31] E. Parada-Cabaleiro, A. Batliner, and B. Schuller, "A diplomatic edition of il Lauro Secco: Ground truth for OMR of white mensural notation," in *Proc. of the International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, 2019, pp. 557–564.

[32] A. Kirkman, "Review: The Josquin research project by Jesse Rodin and Craig Sapp," *Journal of the American Musicological Society*, vol. 68, no. 2, pp. 455–465, 2015.

[33] P. van Kranenburg and G. Maessen, "Comparing offertory melodies of five Medieval Christian chant traditions," in *Proc. of the International Society for Music Information Retrieval Conference*. Suzhou, China: ISMIR, 2017, pp. 204–210.

[34] R. de Valk and T. Weyde, "Bringing 'musicque into the tableture': Machine-learning models for polyphonic transcription of 16th-century lute tablature," *Early Music*, vol. 43, no. 4, pp. 563–576, 2015.

[35] M. Müller, *Fundamentals of music processing*. Cham, Switzerland: Springer Verlag, 2015.

[36] A. Aljanaki and M. Soleymani, "A data-driven approach to mid-level perceptual musical feature modeling," in *Proc. of the International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 615–621.

[37] D. C. Corrêa and F. A. Rodrigues, "A survey on symbolic data-based music genre classification," *Expert Systems with Applications*, vol. 60, pp. 190–210, 2016.

[38] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The Munich versatile and fast open-source audio feature extractor," in *Proc. of ACM Multimedia*. ACM, 2010, pp. 1459–1462.

[39] M. B. Brown and A. B. Forsythe, "372: The ANOVA and multiple comparisons for data with heterogeneous variances," *Biometrics*, pp. 719–724, 1974.

[40] R. L. Wasserstein and N. A. Lazar, "The ASA's statement on p-values: Context, process, and purpose," *The American Statistician*, vol. 70, pp. 129–133, 2016.

[41] D. Lakens, "Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs," *Frontiers in Psychology*, vol. 4, 2013.

[42] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society*, vol. 57, pp. 289–300, 1995.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[44] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. of the {USENIX} Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[45] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[46] H. Verma and J. Thickstun, "Convolutional composer classification," in *Proc. of the International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, 2019, pp. 549–556.