Pattern Recognition Letters
journal homepage: www.elsevier.com

Handwritten Music Recognition for Mensural Notation with Convolutional Recurrent Neural Networks

Jorge Calvo-Zaragoza[a,**], Alejandro H. Toselli[b], Enrique Vidal[b]

[a]*Dept. of Software and Computing Systems, University of Alicante, 03690 Alicante, Spain*
[b]*PRHLT Research Center, Universitat Politècnica de València, 46022 Valencia, Spain*

ABSTRACT

Optical Music Recognition is the technology that allows computers to read music notation, which is also referred to as Handwritten Music Recognition when it is applied over handwritten notation. This technology aims at efficiently transcribing written music into a representation that can be further processed by a computer. This is of special interest to transcribe the large amount of music written in early notations, such as the Mensural notation, since they represent largely unexplored heritage for the musicological community. Traditional approaches to this problem are based on complex strategies with many explicit rules that only work for one particular type of manuscript. Machine learning approaches offer the promise of generalizable solutions, based on learning from just labelled examples. However, previous research has not achieved sufficiently acceptable results for handwritten Mensural notation. In this work we propose the use of deep neural networks, namely convolutional recurrent neural networks, which have proved effective in other similar domains such as handwritten text recognition. Our experimental results achieve, for the first time, recognition results that can be considered effective for transcribing handwritten Mensural notation, decreasing the symbol-level error rate of previous approaches from 25.7 % to 7.0 %.

1. Introduction

Anyone with average knowledge in art may name relevant Renaissance painters or writers. However, it is not the same if asked about music composers. This shortfall is mainly caused by the scarcity of transcribed musical pieces of that time, despite the huge amount of written manuscript sources. This is particularly the case for early music written in the so called *Mensural notation*.

The framework in which the musical pieces of that era were developed fostered, among other things, the massive use of handwritten copies. As a result there are millions of music documents (hand)written in this notation. While large quantities of them have recently been scanned into digital images, only a few hundreds Mensural notation documents have been transcribed and can therefore be effectively used by scholars and general public. Clearly, to make this huge amount of sources available and useful, accurate musical transcripts are required. That is why there is a need for developing systems that are capable of recognizing handwritten music, i.e. Handwritten Music Recognition (HMR) systems.

Several attempts have been made to develop HMR systems in the past decades. Traditional approaches focused on exploiting domain knowledge. However, handwritten music documents (and their trancripts) are very heterogeneous and this type of heuristic systems do not generalize well. Consequently, it is common that new systems need to be built from scratch for each type of manuscripts and, moreover, they seldom reach the required level of usefulness.

In recent years, the paradigm is shifting towards machine learning techniques, which allow for the required generalization as long as adequate training data are available. So far, however, the success of these techniques has only been demonstrated for parts of the whole HMR task, such as the removal of staff lines (a traditional image pre-process in this domain) or the classification of isolated musical elements. More recently, holistic approaches based on machine learning have also been proposed.

**Corresponding author: Tel.: (+34) 963 878 174 Ext. 78174
e-mail: jcalvo@prhlt.upv.es (Jorge Calvo-Zaragoza)

While these approaches provide a well-principled framework and promising results, their accuracy still fall short of what is considered sufficiently successful.

In this paper we further develop the machine learning, holistic paradigm and propose a system based on deep Convolutional and Recurrent Neural Networks, which do prove sufficiently successful for Mensural notation HMR. Specifically, in our experiments we improve the results achieved to date with holistic approaches from 26% to 7% error at the symbol level.

The rest of the paper is organized as follows: we overview related works in Section 2; the neural framework is described in Section 3; the experiments are presented in Section 4; and the paper is concluded in Section 5.

2. Related works

The term Optical Music Recognition (OMR) is rather general, because the task itself depends on several factors such as the notation type (modern Western, Mensural, Neumatic, etc.) or the engraving mechanism (handwritten or printed).

It is true, however, that there has been a general OMR framework to address the recognition of music notation through a series of independent stages that work on different parts of the problem (Wen et al., 2015). Given that music notation hardly has what we might consider low-level entities, like phonemes in speech or characters in text, but rather isolated music symbols, most previous approaches consider by default that symbol segmentation should be an initial step.

Nevertheless, symbol segmentation is often difficult, especially in the case of images of ancient handwritten music. It becomes particularly difficult to distinguish between relevant small elements from noise and other artifacts caused by document preservation problems and possible lack of image quality. Recently, the early stages of the process have been reformulated as object detection tasks (Everingham et al., 2015), with the aim of by-passing some of the stages of the traditional workflow. The state of the art for object detection considers the use of region-based deep neural models (Ren et al., 2015; Dai et al., 2016). Pacha et al. (2018) provided a baseline for direct music-object detection in music score images, experimenting with several models and corpora of different typology.

Conversely, in this paper, we study a holistic approach for HMR using deep neural networks. In this regard, our model performs the complete recognition of musical notation from an image, yielding directly the sequence of music symbols present therein as output. Unlike other recent works (Hajič et al., 2018; Baró et al., 2019), this prevents the need of the training set to be annotated at the symbol-level position and post-process strategies that convert the individually detected elements to the actual music notation .

Concerning this formulation, Pugin (2006) already proposed a holistic approach for printed Mensural notation using Hidden Markov Models (HMM). This approach was recently extended to handwritten sources by using a more appropriate set of features (Calvo-Zaragoza et al., 2016), and further improved by considering discriminative training techniques (Calvo-Zaragoza et al., 2017), as well as hybridization with neural networks (Calvo-Zaragoza et al., 2019).

However, although HMMs represent models that fit perfectly well with the task at issue, other tasks of a similar nature, like speech recognition or handwritten text recognition, have experienced a leap in performance using deep neural networks (Amodei et al., 2016; Shi et al., 2017). This is why in this work we study the use of a deep neural network model for holistic end-to-end HMR. A holistic approach has been explored before but restricted to synthetically-rendered music notation (van der Wel and Ullrich, 2017; Calvo-Zaragoza and Rizo, 2018). Therefore, this is the first work that uses deep neural networks to deal with handwritten notation in old documents.

3. Framework

We follow the holistic approaches to HMR discussed in Sec. 2. The most important novelty here is the use of CRNNs to model the posterior probability of generating output symbols, given an input image. As in previous works, input images are assumed to be single staff-sections, which have been previously detected using well-known simple and robust techniques like that of Cardoso et al. (2009).

A CRNN is composed of one block of *convolutional* layers followed by another block of *recurrent* layers (Shi et al., 2017). Each convolutional layer is usually followed by a max-pooling layer to reduce the dimensionality of its output. The convolutional block is in charge of extracting relevant image features and the recurrent layers interpret these features in terms of sequences of output musical symbols. In this work, the recurrent layers are networks of special "neurons" called "Long Short Term Memory" (LSTM) units, arranged into the so called "Bidirectional LSTM" architecture (BLSTM) (Graves, 2008).

In our previous works, mainly based on HMMs, the system input was a sequence of feature vectors, extracted from the image by means of general feature extraction methods, assumedly adequate for all HMR tasks. In contrast, here no handcrafted feature extraction process is necessary, because the layers of the convolutional block are automatically trained from the task-specific training data to implicitly extract the most adequate features for the images of this task (Zeiler and Fergus, 2014). Moreover, rather than a single, gray-level image, a multi-channel (RGB) image can be directly used as input. In any case, the unit activations in the last convolutional/max-pooling layer can be seen as a sequence of feature vectors representing the input image, $\mathbf{x}$. They can also be seen as linearly downscaled versions of $\mathbf{x}$. Let W be the horizontal size of $\mathbf{x}$, i.e., the width of the input image. The width of the resulting "feature images" will be $J = \gamma W$, where $\gamma \leq 1$ is defined by the max-pooling parameters.

The convolutional block produces as many feature images as the number of filters set in the last layer. All these images are concatenated to form a single feature image, which is fed to the first BLSTM layer. Then, the unit activations of the last recurrent layer are considered estimates of the posterior probabilities per frame:

$$P(\sigma \mid \mathbf{x}, j), \quad 1 \leq j \leq J, \quad \sigma \in \Sigma' \stackrel{\text{def}}{=} \Sigma \cup \{\epsilon\} \tag{1}$$

where Σ is the set of music symbols and ϵ is a special *"non-character"* symbol, needed for images that contain two or more

consecutive instances of the same musical symbol (Graves, 2008). $P(\sigma \mid \mathbf{x}, j)$ is often referred to as the symbol *posteriorgram* of $\mathbf{x}$.

3.1. CRNN training

Convolutional neural networks can be straightforwardly trained through gradient descent using the well-known *Back Propagation* (BP) algorithm. BLSTM networks can be trained similarly by means of a version of BP known as *Back Propagation Through Time* (BPTT) (Williams and Zipser, 1995). Therefore both the convolutional and recurrent blocks of a CRNN can be jointly and uniformly trained, essentially through BP/BPTT.

As it is, the conventional BPTT process requires the information about which symbol must be predicted in each output frame. Nevertheless, a usual HMR training set only provides, for each staff image, its corresponding target transcript into musical symbols, without any kind of explicit information about the framewise location of the symbols. Opportunely enough, it has been shown that the BLSTM layers can be conveniently trained without this information by using the so called "Connectionist Temporal Classification" (CTC) loss function (Graves et al., 2006). The resulting CTC training procedure is a form of Expectation-Maximization, similar to the backward-forward algorithm used for HMM training (Rabiner and Juang, 1993). It is often claimed that the use of a non-character symbol becomes essential for adequate CTC training (Graves et al., 2006).

In order to reduce overfitting, we apply Dropout (Srivastava et al., 2014) during each gradient descent iteration, by disabling a set of units selected at random. In this work, this applied only to units of the recurrent block.

3.2. Statistical language modeling

Music notation exhibits regularities or constraints that, despite being extremely difficult to model in their totality, can be exploited to some extent to improve recognition accuracy. These regularities are often referred to as language constraints.

The CRNN implicitly models these constraints, given that it is trained to estimate the symbol posteriorgram $P(\sigma \mid \mathbf{x}, j)$, which is closely related (see below) to the posterior probability $P(\mathbf{s} \mid \mathbf{x})$, where $\mathbf{s} \in \Sigma^\star$ is a transcript of this image into a sequence of musical symbols. However, in other domains related with HMR, such as handwritten text recognition and automatic speech recognition, it has been clearly shown that the use of an explicit, independently trained language model (LM) can significantly improve recognition accuracy (Bluche, 2015).

Therefore, we follow this idea and explore the impact of using such a kind of LM in the HMR task. Specifically we resort to N-gram models. An N-gram model assumes a local-context simplification of the probability of a sequence $\mathbf{s} = s_1 \ldots s_m$ as[1]:

$$\begin{aligned} P(\mathbf{s}) &= P(s_1) \prod_{i=2}^{m} P(s_i \mid s_1 \ldots s_{i-1}) \\ &\approx \prod_{i=1}^{m} P(s_i \mid s_{i-N+1} \ldots s_{i-1}) \end{aligned} \tag{2}$$

where $P(s_i|s_{i-N+1} \ldots s_{i-1})$ denotes the probability of finding s_i after $s_{i-N+1} \ldots s_{i-1}$. These probabilities are the parameters of the N-gram model, which are easily estimated using the training set transcripts (Vidal et al., 2005).

Given the limited amount of training data, many events might not appear in the training set. In order to generalize better, the smoothing strategy proposed by Kneser and Ney (1995) is used, so that $P(s) > 0 \; \forall s \in \Sigma^\star$.

3.3. Recognition or decoding

Formally, we are given an input image $\mathbf{x}$, which has to be recognized or "decoded" into a most likely music symbol sequence, $\hat{\mathbf{s}} \in \Sigma^\star$:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s} \in \Sigma^*} P(\mathbf{s} \mid \mathbf{x}) \tag{3}$$

Without any explicit LM, Eq. (3) is straightforwardly solved by local optimization. To this end, first an optimal symbol is computed for each posteriorgram position j:

$$\hat{\sigma}_j = \arg\max_{\sigma \in \Sigma'} P(\sigma \mid \mathbf{x}, j) \quad 1 \le j \le J \tag{4}$$

Then an approximately optimal output sequence is obtained as:

$$\hat{\mathbf{s}} = \hat{s}_1 \ldots \hat{s}_m \approx \mathcal{F}(\hat{\sigma}_1 \ldots \hat{\sigma}_J) \quad m \le J \tag{5}$$

where $\mathcal{F} : \Sigma'^J \rightarrow \Sigma^m$ is a function which first merges all the consecutive characters such that $\hat{\sigma}_j = \hat{\sigma}_{j-1}$ and then deletes all the non-character symbols ($\sigma_j = \epsilon$) (Graves et al., 2006).

To use a LM, first Eq. (3) is rewritten as $\hat{\mathbf{s}} = \arg\max_{\mathbf{s} \in \Sigma^\star} P(\mathbf{s})\, p(\mathbf{x} \mid \mathbf{s})$, where $P(\mathbf{s})$ is the LM probability, computed as in Eq. (2). Let $\boldsymbol{\sigma} = \sigma_1 \ldots \sigma_J \in \Sigma'^\star$. Assuming $\mathbf{x}$ is conditionally independent of $\mathbf{s}$ given $\boldsymbol{\sigma}$, $p(\mathbf{x} \mid \mathbf{s})$ can be rewritten as:

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{s}) &= \sum_{\boldsymbol{\sigma}} p(\mathbf{x}, \boldsymbol{\sigma} \mid \mathbf{s}) = \sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma} \mid \mathbf{s}) p(\mathbf{x} \mid \boldsymbol{\sigma}, \mathbf{s}) \\ &= \sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma} \mid \mathbf{s}) p(\mathbf{x} \mid \boldsymbol{\sigma}) \end{aligned} \tag{6}$$

$P(\boldsymbol{\sigma} \mid \mathbf{s})$ can be considered uniform for all $\boldsymbol{\sigma}$ such that $\mathbf{s} = \mathcal{F}(\boldsymbol{\sigma})$ and null for all the other $\boldsymbol{\sigma}$. Therefore,

$$p(\mathbf{x} \mid \mathbf{s}) \propto \sum_{\boldsymbol{\sigma}: \mathcal{F}(\boldsymbol{\sigma}) = \mathbf{s}} p(\mathbf{x} \mid \boldsymbol{\sigma}) \approx \max_{\boldsymbol{\sigma}: \mathcal{F}(\boldsymbol{\sigma}) = \mathbf{s}} p(\mathbf{x} \mid \boldsymbol{\sigma}) \tag{7}$$

Finally, from Eq. (3) and Eq. (7):

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s})\, p(\mathbf{x} \mid \mathbf{s}) \approx \arg\max_{\mathbf{s}} P(\mathbf{s}) \max_{\boldsymbol{\sigma}: \mathcal{F}(\boldsymbol{\sigma}) = \mathbf{s}} p(\mathbf{x} \mid \boldsymbol{\sigma}) \tag{8}$$

$p(\mathbf{x} \mid \boldsymbol{\sigma})$ is obtained from the posteriorgram of $\mathbf{x}$ as follows:

$$p(\mathbf{x} \mid \boldsymbol{\sigma}) \approx p(\mathbf{x}) \prod_{j=1}^{J} \frac{P(\sigma_j \mid \mathbf{x}, j)}{P(\sigma_j)^\kappa} \tag{9}$$

[1] For the sake of notation simplicity, for any sequence $\mathbf{z}$ if $j < 1$, $P(z_k \mid z_j \ldots z_{k-1})$ is assumed to denote $P(z_k \mid z_1 \ldots z_{k-1})$. If $j = 1$, it is just $P(z_1 \mid \lambda) \equiv P(z_1)$, where λ is the empty sequence.

where the factor $p(\mathbf{x})$ can be ignored in Eq. (8) since it does not depend on $\boldsymbol{\sigma}$. The symbol priors $P(\sigma)$, $\sigma \in \Sigma'$, can be straightforwardly estimated from training data and κ is a meta-parameter to be tuned empirically (Bluche, 2015).

Clearly, Eq. (8) can not longer be solved by local optimization, but sufficiently accurate solutions can be obtained using the Viterbi algorithm (which is sometimes referred to as "token passing" or "max-product belief propagation").

In this work the N-gram model is represented as a finite-state transducer whose edges are weighted with the product of the N-gram probabilities (the factors in Eq. (2)) and the optical model probabilities (the factors of Eq. (9), without the term $p(\mathbf{x})$). Finally, Eq. (8) is solved using the Viterbi beam search decoder implemented in the Kaldi toolkit (Povey et al., 2011).

To summarize, Fig. 1 illustrates the whole pipeline to decode the input image $\mathbf{x}$ into an (approximately) optimal sequence of music symbols $\hat{\mathbf{s}}$ using a CRNN and an LM.

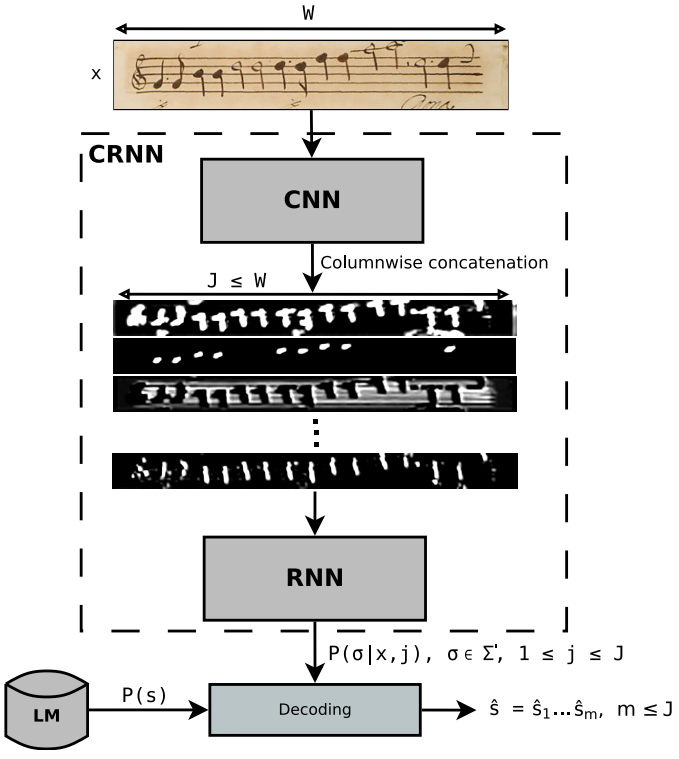


Figure 1. General overview of the HMR framework proposed in this work, from an input image depicting a staff-section region to decoding it into a sequence of music symbols by means of a CRNN and an LM.

4. Experiments

In this section, the experiments carried out to validate the goodness of the proposed framework will be presented. The dataset used and the evaluation protocol are also described before the actual results.

4.1. Corpus

We consider CAPITAN corpus (Calvo-Zaragoza et al., 2017), which contains a complete 96−page manuscript of the 17th century corresponding to a *missa* (sacred music). Each page was already segmented into staff-section images following a semi-automatic procedure.

As in almost any music notation, the meaning of most musical symbols relies on two geometrical informations: *shape* and *height* (vertical position of the symbol in the staff), which mostly indicate the duration and pitch, respectively. In the case of the considered notation, this duality is more general because even symbols that do not denote any sound (such as *rests*) may also appear at different heights, which was useful for reading the music when many rests appear consecutively. In this regard, each possible combination of shape and height is considered here as a unique symbol, which leads to a vocabulary of 183 different symbols.

A standard partition into training, validation, and test samples was already established, which allows us to fairly compare our performance with previous results over the same dataset. A summary of the characteristics of the dataset as regards to this partition is given in Table 1.

Table 1. Partition of the Capitan dataset, reporting the number of staves, the number of different music symbols (or "vocabulary") and the number of running symbols.

	Training	Validation	Test
Staves	462	57	57
Different symbols	176	123	115
Running symbols	10 323	1 286	1 254

4.2. Evaluation protocol

Taking into account the different elements of the HMR task, we consider several metrics to measure the recognition performance, namely:

- *Diplomatic Symbol Error Rate* (SER): computed as the average number of elementary editing operations needed to produce a reference (correctly transcribed) symbol sequence from the recognized symbol sequence.
- *Glyph Error Rate* (GER): as in SER but only taking into account the shape of the symbols, ignoring the height component (where any).
- *Height Error Rate* (HER): as in SER but only taking into account the height of the symbol. Those symbols that have no height are grouped into the same one.

4.3. Image pre-processing

Although the CRNN is able to learn to extract features from the examples using the convolutional layers, it is often convenient to perform simple normalization processes to ensure that the staff-section images are always presented in a similar way. The following steps are applied in this work:

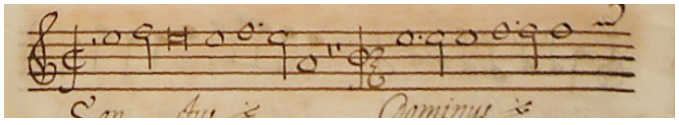
1. Skew correction: the image skew is computed and corrected so that the staff remains aligned with the horizontal axis. We use the staff-line detection algorithm proposed by Cardoso et al. (2009) which capitalizes on the excellent reference provided by the staff lines themselves.

2. Staff location: to ensure that the staff section to be processed is well framed, we force the middle line of the staff to be in the centre of the image. In addition, the image is cropped so that it has a fixed height of 1.5 times the distance between the first and last line of the staff. The average height of the resulting staff images is 256 pixels.

3. Height normalization: the recognition methods to be applied require that each image column be of a fixed height. Therefore, the image is rescaled to a fixed height, without changing the aspect ratio. The specific height will be empirically studied.

Note that neither the staff section separation nor these normalization steps remove the possible accompanying text (lyrics), which is just considered "noise" for the music notation recognition. Fig. 2b shows the result of applying this preprocessing stage to the staff image depicted in Fig. 2a.

(a) Original staff sample.

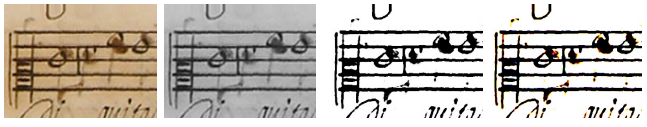


(b) Skew correction and staff normalization.

Figure 2. Pre-processing steps applied to an original staff sample for the HMR task.

Once the staff-region image has been normalized, there are several ways to present the image. In particular, we consider the following image transformations (illustrated in Fig.-3):

- Color (Fig. 3a): the staff section is used as it is (RGB), without any image transformation.
- Grayscale (Fig. 3b): the staff section is transformed into grayscale mode.
- Binarization (Fig. 3c): the staff section is binarized using Sauvola's method (Sauvola and Pietikäinen, 2000).
- Channel-wise binarization (Fig. 3d): the same as the previous case but considering the binarization independently in each channel.



(a) (b) (c) (d)

Figure 3. Examples of the pre-processing transformations applied to the input staff sections: (a) Original RGB image, (b) Grayscale, (c) Binarization, (d) Channel-wise binarization. This figure must be seen in color.

In the experiments, we will analyze the impact of the different image transformations in the recognition performance.

4.4. CRNN configuration tuning

We study empirically the impact of different CRNN topologies or architectures. In order to reduce the search space, we have restricted ourselves to a template CRNN, described in Table 2. It consists of 4 convolutional layers with *Leaky ReLU* activation (Maas et al., 2013) and max-pooling down-sampling, and 2 BLSTM recurrent layers. The height of the input image is parameterized, and rescaled keeping the aspect ratio, and the input layer is configured with as many channels as needed for each image transformation considered (see Sec. 4.3). This template defines a set of CRNN topologies, which show to be successful in the HTR field (Shi et al., 2017).

Table 2. Template of the CRNN used in this work, consisting of 4 convolutional layers and 2 recurrent layers. The model accepts a variable-width image, whose height is parametrized. The number of channels depends on the image pre-process considered. Notation: Input($h\times w\times c$) means an input image of height h, width w and c channels; Conv($n, h\times w$) denotes a convolution operator of n filters and kernel size of $h\times w$; MaxPooling($h\times w$) represents a down-sampling operation of the dominating value within a window of size ($h\times w$); BLSTM(n) means a bi-directional Long Short-Term Memory unit of n neurons; Dropout(p) represents a dropout operation with a ratio of p per iteration; Dense(n) denotes a dense layer of n neurons; and Softmax() represents the *softmax* activation function. Σ denotes the alphabet of musical symbols considered.

Input($h \times W \times C$)
Conv($64, 5 \times 5$), MaxPooling($2 \times p_1$)
Conv($64, 5 \times 5$), MaxPooling($2 \times p_2$)
Conv($128, 3 \times 3$), MaxPooling($2 \times p_3$)
Conv($128, 3 \times 3$), MaxPooling($2 \times p_4$)
BLSTM(256), Dropout(0.5)
BLSTM(256), Dropout(0.5)
Dense($\lvert\Sigma'\rvert$), Softmax()

In addition, in the experiments, we study the impact of the following hyper-parameters:

- Height of the image (h): using the image at its original size may represent a high complexity for the learning process — especially given the limited number of training samples. Therefore, a rescaling process of the image is considered, at fixed heights of 32, 64, and 128 pixels, and keeping the aspect ratio. Given that the original staff-section samples depict an average height of 250 pixels, these parameters represent an approximate average scale factor of 0.125, 0.25, and 0.5, respectively.

- Horizontal pooling (p_1, p_2, p_3, p_4): an important aspect is the amount of frames that exist at the output of the convolutional block. Note that, because of the CTC loss function operation, during training the recurrent block must provide at least twice as many frames as the expected sequence length. The parameterization of the horizontal pooling therefore represents a balance between complexity and flexibility for the recurrent block. That is why we consider the values (2,2,2,1), (2,2,1,1), and (2,1,1,1), to cover many options for this trade-off.

Table 3. Symbol error rate (SER, in %) with respect to hyper-parameters over the validation set. Best value and parameters are typeset in boldface. Notation: h indicates the fixed image height; (p_1, p_2, p_3, p_4) denote the consecutive horizontal pooling values of the convolutional block; C, G, B, and Ch.B represent the color, grayscale, binarization, and channel-wise binarization image transformations, respectively.

		Image transformation			
h	(p_1, p_2, p_3, p_4)	C	**G**	B	Ch.B
32	2,2,2,1	-	-	-	-
	2,2,1,1	23.5	23.7	25.3	24.0
	2,1,1,1	13.9	14.8	17.7	16.7
64	2,2,2,1	10.9	12.5	12.0	10.3
	2,2,1,1	6.1	6.6	7.0	6.2
	2,1,1,1	5.2	5.1	7.3	4.9
128	2,2,2,1	6.1	5.4	5.9	5.4
	2,2,1,1	4.7	**4.5**	4.7	4.7
	2,1,1,1	4.9	5.3	4.7	4.8

Table 3 shows results considering several combinations of the image transformations and hyper-parameters mentioned above. Note that this experiment is performed only over the validation partition. For simplicity, only the most general metric (SER) is considered here.

An initial remark is that the results of each row remain quite similar, which indicates that the transformation applied to the images is not as relevant as other parameters. The fixed height of the images is the most relevant parameter, resulting in differences between 10 % and 20 % of SER. Within the block that represents each fixed height, we observe that the way of doing the consecutive pooling operations also has an impact on the results, yet to a lesser extent.

According to these results, for the following recognition experiments we adopt a configuration consisting of a single input chanel (original grayscale image) normalized to a height of 128 pixels, and a series of horizontal pooling operations of (2,2,1,1).

4.5. Recognition results

We carried out the final evaluation over the test set for increasing N-gram orders using the best configuration determined in Sec. 4.4 with the validation set.

In addition, the accuracy of the considered approach is compared with previous work over the same corpus. Specifically: HMMs both trained with the classic Maximum Likelihood (ML) estimation (Calvo-Zaragoza et al., 2016) and with Discriminative Training (DT) (Calvo-Zaragoza et al., 2017), as well as HMMs hybridized with Multi-Layer Perceptron (MLP) models (Calvo-Zaragoza et al., 2019). In these cases, we directly consider the results with the best N-gram estimation found in the aforementioned references. For the sake of comparison, we also report the performance of some previous research on recognition of music notation, such as Aruspix system (Pugin, 2006) and the neural sequence-to-sequence approach (seq2seq) described in the work of van der Wel and Ullrich (2017). It is really important to emphasize that this comparison is not totally fair, as these methods were not designed to work with handwritten notation, which is the object of study here. However, including them might help to put into context our work.

All these works have been implemented following the details provided in the corresponding publication, except in the case of Aruspix[2], and evaluated under the same experimental conditions considered for the CRNN.

Table 4. Summary of final results for different HMR approaches, including those achieved in this work in the last two rows.

Method	SER	GER	HER
HMM-GMM-ML + 4-gram	46.2	41.2	34.9
HMM-GMM-DT + 3-gram	40.4	35.2	28.2
HMM-MLP + 3-gram	25.7	22.4	18.7
Aruspix	94.5	93.0	94.1
seq2seq	27.8	24.5	26.5
CRNN-CTC	7.3	5.8	5.1
CRNN-CTC + 3-gram	**7.0**	**5.6**	**4.9**

The final recognition results are shown in Table 4. An inspection of the reported figures reveals two relevant conclusions. On one end, the use of CRNN-CTC drastically improves all the results obtained previously with HMM, even those that were attained by hybridizing HMM with MLP. In this case, the error is reduced from around 26 % to 7 %, at the full symbol level. This implies that the approach proposed in this work achieves, for the first time, recognition results that can be considered effective for handwritten music notation in old documents. The two compared approaches that were not designed for HMR report an unalike performance. Aruspix misclassifies almost all symbols, thus yielding an error close to 100 %. The neural seq2seq approach does correctly recognize much more symbols, and provides a fair performance that is close to that based on HMMs hybridized with MLP, yet quite far from the performance attained by the proposed CRNN-CTC approach.

On the other end, it is observed that even in this improved scenario, when the error is already relatively low, adding an N-gram statistical LM can be beneficial, slightly decreasing the error up to 0.3 %. It should be noted that this statistical model was estimated from a quite limited corpus (the training partition only includes 462 samples) so it would be interesting to verify its impact with a greater number of samples.

As regards to the dual nature of the symbols, it follows from GER and HER figures that both the height and the shape symbol components contribute almost equally to the accuracy, as happened in previous approaches. The HER is systematically lower in all cases, which is not surprising considering that there are 35 different shapes but only 16 height positions.

5. Conclusions

In this paper we consider a neural approach based on CRNN, for the task of HMR in Mensural notation. Unlike traditional

[2]Aruspix software is available at `http://www.aruspix.net/` (last accessed 22-07-2019).

approaches, the recognition task is modeled in an end-to-end way, for which the training stage only needs pairs of images and their corresponding transcripts in the form of music symbol sequences.

In our experiments over a 17th century manuscript, we empirically evaluate which hyper-parameters are most suitable, such as the neural network configuration, the height of the input, or the pre-processing of the image. These experiments confirm that the selection of these hyper-parameters might have a great impact on the performance of the model. With the best of these configurations, the obtained recognition results improve considerably compared to previous holistic approaches based on HMM, decreasing the symbol-level error rate from 25.7 % to 7.0 %.

Additionally, we have considered the use of statistical LMs to lead the recognition towards those hypotheses that are most promising *a priori*. Given that the recurrent layers of the CRNN implicitly provide the same kind of contextual modelling by themselves, we have observed that the improvements that can be obtained are limited — at least when the statistical LM is estimated with the same samples that are used to train the CRNN. As future work, it might be interesting to measure the impact of a LM when it is estimated from data outside the training set, which is especially interesting because it only needs series of transcripts (without their associated images).

Another interesting avenue for future research is to model the double shape-height nature of symbols, as it is one of the main features that distinguish music notation from other similar domains such as text. In this work we considered that each combination of these two components must be understood by the system as a totally new independent symbol. Nevertheless, all notes of the same shape share many features, and this information is not being used to improve recognition performance. Also, a more convenient estimation can possibly be obtained from independent shape and height statistical LMs.

Furthermore, given the limited amount of data and the relatively large number of symbols, data augmentation represents an interesting framework to consider to make the neural model more robust. However, it should be noted that data augmentation for music staves must go beyond simple blurring, rotation, and scaling. In particular, staff lines are elements that should remain basically similar in all images, and only musical symbols should be altered.

Acknowledgments

First author thanks the support from the Spanish Ministry "HISPAMUS" project (TIN2017-86576-R), partially funded by the EU. The other authors were supported by the European Union's H2020 grant "Recognition and Enrichment of Archival Documents" (Ref. 674943).

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al., 2016. Deep speech 2: End-to-end speech recognition in english and mandarin, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, pp. 173–182.

Baró, A., Riba, P., Calvo-Zaragoza, J., Fornés, A., 2019. From optical music recognition to handwritten music recognition: A baseline. Pattern Recognition Letters 123, 1–8.

Bluche, T., 2015. Deep Neural Networks for Large Vocabulary Handwritten Text Recognition. Ph.D. thesis. Ecole Doctorale Informatique de Paris-Sud - Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur. Discipline : Informatique.

Calvo-Zaragoza, J., Rizo, D., 2018. End-to-end neural optical music recognition of monophonic scores. Applied Sciences 8, 606–629.

Calvo-Zaragoza, J., Toselli, A.H., Vidal, E., 2016. Early handwritten music recognition with hidden markov models, in: 15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016, pp. 319–324.

Calvo-Zaragoza, J., Toselli, A.H., Vidal, E., 2017. Handwritten music recognition for mensural notation: Formulation, data and baseline results, in: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, pp. 1081–1086.

Calvo-Zaragoza, J., Toselli, A.H., Vidal, E., 2019. Hybrid hidden markov models and artificial neural networks for handwritten music recognition in mensural notation. Pattern Analysis and Applications doi:10.1007/s10044-019-00807-1.

Cardoso, J.S., Capela, A., Rebelo, A., Guedes, C., Pinto, J., 2009. Staff detection with stable paths. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1134–1139.

Dai, J., Li, Y., He, K., Sun, J., 2016. R-FCN: object detection via region-based fully convolutional networks, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, Barcelona, Spain, pp. 379–387.

Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision 111, 98–136.

Graves, A., 2008. Supervised sequence labelling with recurrent neural networks. Ph.D. thesis. Technical University Munich.

Graves, A., Fernández, S., Gomez, F., Schmidhuber, J., 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, New York, NY, USA. pp. 369–376.

Hajič, J.j., Dorfer, M., Widmer, G., Pecina, P., 2018. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets, in: 19th International Society for Music Information Retrieval Conference, Paris, France. pp. 225–232.

Kneser, R., Ney, H., 1995. Improved backing-off for m-gram language modeling, in: 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, pp. 181–184.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models, in: ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Atlanta, Georgia, USA.

Pacha, A., Hajič, J., Calvo-Zaragoza, J., 2018. A baseline for general music object detection with deep learning. Applied Sciences 8, 1488.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., 2011. The Kaldi speech recognition toolkit, in: Workshop on Automatic Speech Recognition and Understanding (ASRU2011), pp. 1–4.

Pugin, L., 2006. Optical music recognitoin of early typographic prints using hidden markov models, in: Proceedings of the 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October, pp. 53–56.

Rabiner, L., Juang, B.H., 1993. Fundamentals of speech recognition. Prentice hall.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, December 7-12, Montreal, Quebec, Canada, pp. 91–99.

Sauvola, J., Pietikäinen, M., 2000. Adaptive document image binarization. Pattern Recognition 33, 225–236.

Shi, B., Bai, X., Yao, C., 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2298–2304.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15, 1929–1958.

Vidal, E., Thollard, F., De La Higuera, C., Casacuberta, F., Carrasco, R.C., 2005. Probabilistic finite-state machines-part ii. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1026–1039.

van der Wel, E., Ullrich, K., 2017. Optical music recognition with convolutional sequence-to-sequence models, in: Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, October 23-27, pp. 731–737.

Wen, C., Rebelo, A., Zhang, J., Cardoso, J.S., 2015. A new optical music recognition system based on combined neural network. Pattern Recognition Letters 58, 1–7.

Williams, R.J., Zipser, D., 1995. Backpropagation, L. Erlbaum Associates Inc., Hillsdale, NJ, USA. chapter Gradient-based Learning Algorithms for Recurrent Networks and Their Computational Complexity, pp. 433–486.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, in: Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, September 6-12, Part I, pp. 818–833.