A FEW-SHOT NEURAL APPROACH FOR LAYOUT ANALYSIS OF MUSIC SCORE IMAGES

Francisco J. Castellanos[1] **Antonio Javier Gallego**[1] **Ichiro Fujinaga**[2]
[1] University Institute for Computing Research, University of Alicante, Spain
[2] Schulich School of Music, McGill University, Montreal, Canada
{fcastellanos, jgallego}@dlsi.ua.es, ichiro.fujinaga@mcgill.ca

ABSTRACT

Optical Music Recognition (OMR) is a well-established research field focused on the task of reading musical notation from images of music scores. In the standard OMR workflow, layout analysis is a critical component for identifying relevant parts of the image, such as staff lines, text, or notes. State-of-the-art approaches to this task are based on machine learning, which entails having to label a training corpus, an error-prone, laborious, and expensive task that must be performed by experts. In this paper, we propose a novel few-shot strategy for building robust models by utilizing only partial annotations, therefore requiring minimal human effort. Specifically, we introduce a masking layer and an oversampling technique to train models using a small set of annotated patches from the training images. Our proposal enables achieving high performance even with scarce training data, as demonstrated by experiments on four benchmark datasets. The results indicate that this approach achieves performance values comparable to models trained with a fully annotated corpus, but, in this case, requiring the annotation of only between 20% and 39% of this data.

1. INTRODUCTION

Optical Music Recognition (OMR) is a research field dedicated to developing computational methods for transcribing musical notation from document images into digital formats [1]. While this task could be accomplished manually, the vast number and heterogeneity of music documents make this approach tedious, costly, and error-prone. The development of OMR systems has the potential to enhance music heritage accessibility and preservation, as well as enable the application of analysis algorithms to increase knowledge about this cultural legacy.

OMR typically follows a sequential workflow, which divides the transcription process into simpler tasks. The initial task is called Document Image Analysis (DIA), which is itself a research field that studies how to obtain a segmented version of the image by isolating the different layers of interest, such as staves, lyrics, instructions, ornaments, etc [2]. In the literature, multiple strategies can be found to perform this *layout analysis*, ranging from heuristic approaches that exploit specific features of the images to deep learning techniques. Although heuristic approaches achieve high performance in controlled scenarios, these solutions are poorly generalizable. To obtain better and generalizable results, the current trend is to rely on machine learning and, more specifically, on neural network architectures [3].

The application of deep learning in layout analysis has been extensively studied, as evidenced by several state-of-the-art works [4, 5]. However, a major drawback of these methods is the requirement for a large amount of annotated data for their training. This is particularly problematic for the layout analysis of music scores since their high variability in appearance and styles makes necessary the annotation of each new application domain in order to train robust models. Despite the importance of this issue, it has been overlooked in the OMR literature, with domain adaptation being the only explored solution [6]. Nevertheless, this technique also requires full annotations (even if it is from a different domain) and the performance obtained is not good or robust enough, which also makes it an impractical solution.

In this work, we propose a novel few-shot strategy for building robust models for layout analysis by utilizing only partial annotations, therefore requiring minimal human effort. Specifically, we introduce a masking layer and an oversampling technique to train models using a small set of annotated patches from the training images. Our approach aims to drastically reduce the manual workload without compromising performance, making it of particular interest to real-world applications. Experiments on four benchmark datasets indicate that this approach achieves performance comparable to models trained on a fully annotated corpus—but requiring the annotation of only between 20% and 39% of this data depending on the layer—thus making it a highly efficient and effective strategy.

2. RELATED WORK

Traditional OMR workflows relied on a combination of heuristic strategies to perform pixel-wise layout analysis and classify each pixel of the image according to a set

Attribution: F. J. Castellanos, A. J. Gallego, and I. Fujinaga, “A Few-shot Neural Approach for Layout Analysis of Music Score Images”, in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

of categories [2]. A binarization process was commonly applied to simplify the complexity of the image to detect the ink pixels, either using generic approaches [7, 8] or other particular ones proposed for the musical context [9, 10]. The recognition and isolation of the staff and the lyrics were then carried out using also heuristic techniques [11, 12]. From these detected staves, the musical symbols were finally processed, sometimes carrying out another step to remove the staff lines, as can be seen in the review by Dalitz et al. [13] or in more recent works [14–16].

More recently, all these steps were combined by means of machine learning techniques. Calvo-Zaragoza et al. [17] proposed a Convolutional Neural Network (CNN) to directly classify each pixel of the image—performing a pixel-wise layout analysis—which was later improved using a U-net-like architecture—referred to as Selectional Auto-Encoder (SAE)—to more efficiently classify the image by patches [18]. This later work, on which our proposal is based, trained a set of SAE specialized in the detection of each layer of information—staff lines, notes, text, or background.

The main challenge with layout analysis approaches that rely on supervised learning is the large amount of annotated data needed to train the models [19, 20]. This requires the annotation at the pixel level of a reference set of images, which has to be done by hand, so it is not a scalable solution given the high level of detail of these annotations and heterogeneity in music documents. In addition, when this constraint cannot be fulfilled, these learning-based architectures fail to converge to obtain a suitable model for the task at hand.

In the literature, we can find different proposals that seek to alleviate this issue [21], two of the most common being the use of regularization strategies [22] and data augmentation processes [23]. We can also find more specific proposals for cases of remarkable data scarcity, i.e., with a considerably fewer number of annotated training samples. These scenarios are known as *few-shot learning* [24] and typically employ specific neural architectures to estimate the similarity of the data [25]. Some of the most typical examples of these techniques are Siamese Neural Networks [26], Matching Networks [27], Prototypical Networks [28], and Relation Networks [29]. For a comprehensive review of these strategies, the reader is referred to the work by Jadon [30].

Our proposal follows a few-shot learning approach, but instead of using a specific few-shot architecture, a state-of-the-art layout analysis model—the previously described SAE network—is modified to integrate a masking layer that enables training with very little data. This layer is complemented by an oversampling proposal used during the training process to draw samples at random positions around the chunks with annotated data. A mask is applied to these pieces and used by the added layer to avoid processing the non-annotated parts, which will randomly appear in different positions in each iteration, thus forcing the architecture to generalize the learned weights.

In the related literature, masks have been used for different purposes. For example, Medhat et al. [31] proposed the use of binary masks for sound classification to filter out certain frequency bands. It has also been explored for image classification, specifically, Suresh et al. [32] studied the use of masks as a pre-processing task to filter the background of images with hand gestures, making the model focus only on the gestures to be classified. However, as far as we know, masks have not been used either in binarization tasks or for few-shot learning cases, so that the model does not use the unlabeled areas.

3. METHODOLOGY

Our approach aims to build a robust few-shot learning model for layout analysis of music score images that classifies each pixel of an input image into one of the following categories: `staff`, `notes`, `text`, and `background`. In our context, the few-shot scenario can be represented as a manual annotation of a limited number n of portions or patches from a set of images $\mathcal{I}$, with $n \ll N$, where N is the total number of possible patches that could be sequentially extracted without overlapping from $\mathcal{I}$. Therefore, when n is small, less human effort and cost are required to annotate the training set.

Note that labeling only part of the image makes the rest of it uninformative, even if there are ink pixels. In a typical training process, only the annotated patches would be used. However, when the amount of data is limited, this would lead to overfitting of the model. Although data augmentation may help mitigate this problem, in a few-shot learning scenario, it is not very useful due to the little information to be altered.

Our proposal introduces a novel approach to extract a larger—and more varied—number of samples from the scarce labeled information. Specifically, it is proposed to extract random patches around the annotated areas—keeping a minimum $\lambda\%$ of labeled information—to obtain more varied samples, thus generating variations in the position of the elements and their labeling. Since some parts of the extracted patches will fall outside the annotated area, it is proposed to mark those parts with a special label (-1) so that they are not used during training. This approach allows us to control the number of samples to be drawn from the images and get enough variability in the data to train the model, as we will demonstrate in the experiments.

Formally, let $\mathcal{X} \in \mathbb{R}^{w \times h}$ be a collection of patches of size $w \times h$ drawn from the input set of images $\mathcal{I}$, and $\mathcal{Y} \in \{0, 1\}^{w \times h}$ be the corresponding pixel-level annotation matrices extracted from the annotation set $\mathcal{L}^l$ for the layer to be processed $l \in \{\texttt{staff}, \texttt{notes}, \texttt{text}, \texttt{background}\}$, where 1 is used to label the ink of that layer and 0 the rest, either background or information from another layer. Additionally, let $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : \boldsymbol{x}_i \in \mathcal{X}, \boldsymbol{y}_i \in \mathcal{Y}\}_{i=1}^{|\mathcal{S}|}$ represent an annotated collection of data where each datum $\boldsymbol{x}_i$ is related to label $\boldsymbol{y}_i$ by an underlying function $f^l : \mathcal{X} \rightarrow \mathcal{Y}$, that represents the objective function to be learned for each layer l, and for which the SAE state-of-the-art architecture will be used. Note also that $\boldsymbol{x}^*$ will be used to refer to the input patches after applying the mask, which may contain values in the range $[0, 255]$, for the original pixels of the image, but also the value -1 as a mask to mark the parts without annotated information. This mask is therefore applied to the input data in $\mathcal{X}$ and will be used by the masking layer (described below) added to the networks f^l to ignore those parts during the training process.

Algorithm 1 describes the oversampling method proposed to obtain the set $\mathcal{S}$ previously described. This method receives as input the set of images $\mathcal{I}$, the set with the annotated data $\mathcal{L}$, the layer l to be processed, the $\lambda\%$ of minimum patch information, the total *size* of sampling to perform, and the set $\mathcal{M}$ that contains the list of patches annotated with their coordinates in the input images. The algorithm first iterates through the number of patches annotated in $\mathcal{M}$ (`line 3`) and for each one obtains the index j of the image it corresponds to (`line 4`). It then iterates for the number of samples that have to be extracted for that annotated patch (`line 5`) and, for each one, performs the following steps: 1) randomly selects the sample coordinates p using the mask of that patch and taking into account the minimum $\lambda\%$ of annotated pixels allowed (`line 6`); 2) extracts the patch $\boldsymbol{x}$ from $\mathcal{I}_j$ using the coordinates p (`line 7`); 3) applies the mask to set a constant value (-1) in those pixels that are not part of the annotated area (`line 8`); 4) retrieves the layout annotations $\boldsymbol{y}$ for that sample (`line 9`); and 5) both $\boldsymbol{x}^*$ and $\boldsymbol{y}$ are added to the set $\mathcal{S}$. The algorithm repeats this process until reaching the requested *size*, finally returning the set $\mathcal{S}$ obtained.

Algorithm 1 Random masking patches generator.

```
 1: function SampleGeneration($\mathcal{I}, \mathcal{L}, \mathcal{M}, l, \lambda, size$)
 2:     $\mathcal{S} \leftarrow \emptyset$
 3:     for $i \leftarrow 1$ to $|\mathcal{M}^l|$ do
 4:         $j \leftarrow$ getPatchIndex($\mathcal{M}^l_i$)
 5:         for $k \leftarrow 1$ to $\frac{size}{|\mathcal{M}^l|}$ do
 6:             $p \leftarrow$ getRandomPosition($\mathcal{M}^l_i, \lambda$)
 7:             $\boldsymbol{x} \leftarrow$ getWindow($\mathcal{I}_j, p$)
 8:             $\boldsymbol{x}^* \leftarrow$ applyMask($\boldsymbol{x}, \mathcal{M}^l_i, p$)
 9:             $\boldsymbol{y} \leftarrow$ getWindow($\mathcal{L}^l_j, p$)
10:             $\mathcal{S} \leftarrow \mathcal{S} \cup (\boldsymbol{x}^*, \boldsymbol{y})$
11:         end for
12:     end for
13:     return $\mathcal{S}$
14: end function
```

Note that the `getWindow`$(\cdot)$ function may apply additional data augmentation to the sample in order to further increase its variability.

This oversampling process is complemented by the proposal of a masking layer that is added to the network architecture f^l to ignore the pixels that are not annotated. This layer, as indicated in Section 2, has been previously used in other proposals to skip time steps in sequence processes and to mask the background in classification tasks. In this proposal, we adapt it to ignore the parts of the input with this mask and also propagate the mask to the following layers so that the non-annotated parts are not taken into account during the training process. Intuitively, the masking layer acts as a regularizer and data augmentation process. Given that the annotated and non-annotated parts will vary in position and size across iterations, the network is forced to generalize the weights learned during training by having to use different connections of the network and non-annotated pixels will not be used.

4. EXPERIMENTAL SETUP

This section describes the corpora and metrics considered for evaluation and the implementation details of the neural architecture. [1]

4.1 Corpora

For the experiments, we considered the following 4 datasets with manual pixel-wise annotations of 4 layers of information (`staff`, `notes`, `text`, and `background`). Figure 1 shows some examples for each manuscript and Table 1 includes a summary with their details.

- **Ein**: 9 high-resolution scanned pages of neumatic notation belonging to the Einsiedeln, Stiftsbibliothek, Codex 611(89), from 1314. [2]
- **Sal**: A set of 10 high-resolution images of pages from the Salzinnes Antiphonal manuscript (CDM-Hsmu M2149.14), in neumatic notation. It is available in the *Cantus Ultimus* platform. [3]
- **MS73**: Selection of 10 pages of square music notation from the miscellaneous choir book '*Dominican, CDN-Mlr MS Medieval 0073*' from Northern Italy, written between 13th and 15th centuries. This corpus is stored in the McGill Library collection, and it is online available through *Cantus Ultimus*. [4]
- **Cap**: A compilation of mensural notation manuscripts from the 17-18th centuries belonging to the 'Cathedral of Our Lady of the Pillar' in Zaragoza (Spain), introduced for OMR purposes by Calvo-Zaragoza et al. [33]. We use a subset of the corpus, with 10 manually pixel-wise annotated pages.

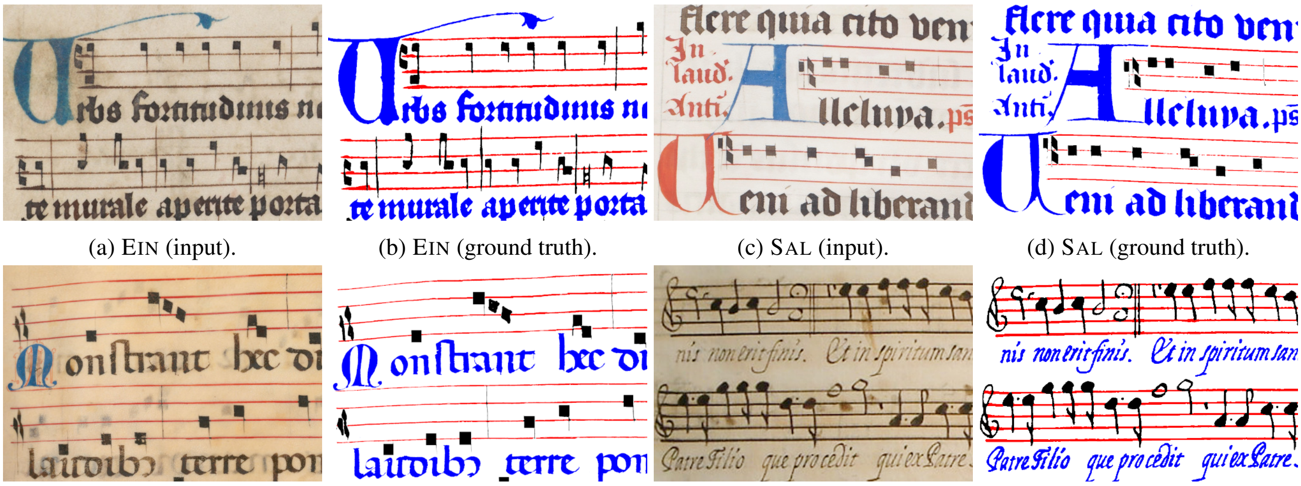
In all the cases, we used 4 images for training, 2 images for validation, and the remaining for testing. After preliminary experiments and also based on previous proposals, we selected a patch size of 256×256 pixels to extract from these images. To be fair and more realistic, we use the same number of samples for the validation set as for the training partition. This is because, in a real case, it would be necessary to annotate the validation partition as well, so it is not fair to use the entire pages to validate the models in a few-shot scenario. This does not apply to the test set, for which we use all available data.

[1] `https://github.com/fjcastellanos/FewShotLayoutAnalysisMusic.git`

[2] `http://www.e-codices.unifr.ch/en/sbe/0611/`

[3] `https://cantus.simssa.ca/manuscript/133/`

[4] `https://cantus.simssa.ca/manuscript/35/`



(a) Ein (input). (b) Ein (ground truth). (c) Sal (input). (d) Sal (ground truth).

(e) MS73 (input). (f) MS73 (ground truth). (g) Cap (input). (h) Cap (ground truth).

Figure 1: Examples of images extracted from the corpora described in Table 1. In the ground truth images: red pixels represent the staff lines annotation, black is used for music symbols, blue for text, and white for the background.

Corpus	# imgs	Resol.	Layers (%)			
			BG	St	No	Te
Ein	9	6 496 × 4 872	87.9	3.5	2.7	5.9
Sal	10	5 847 × 3 818	87.6	2.4	2.5	7.5
MS73	10	6 990 × 4 797	93.4	1.8	1.8	3.0
Cap	10	2 126 × 3 065	85.7	6.6	5.1	2.6

Table 1: Details of the corpora considered including the number of images (# imgs), the average resolution and the proportion of pixels for each layer of interest, with **BG** for background, **St** for staff lines, **No** for notes, and **Te** for text.

4.2 Metrics

To evaluate the performance of our few-shot approach, we resorted to the F-score (F_1) figure of merit to avoid possible biases toward any particular class given the inherent label imbalance in the datasets considered (see Table 1). Assuming a binary classification scenario, this metric is defined as

$$\text{F}_1 = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}, \quad (1)$$

where TP, FP, and FN denote the True Positives, False Positives, and False Negatives, respectively.

Finally, given the non-binary nature of the task at hand, we considered the use of the macro-averaged F-score (F_1^{m}) as the average of the F_1 values computed for each layer. Mathematically, this metric is defined as

$$\text{F}_1^{\text{m}} = \frac{\sum_{l=1}^{|\mathcal{L}|} \text{F}_1^l}{|\mathcal{L}|}, \quad (2)$$

where F_1^l is the F_1 calculated for the layer l assuming a one-versus-all evaluation framework and $|\mathcal{L}|$ represents the total number of layers of information (in our case 4).

4.3 Implementation details

The architecture considered is based on a previous work [18], in which a framework consisting of a series of SAE models—one for each layer to be predicted—was proposed. SAE follows a U-net architecture, in which an image of size $w \times h$ (in our case a 256×256 pixels patch) is given as input, and the output is a matrix of the same size that contains the confidence value of pixels belonging to the layer of interest. In our case, we have four layers to be predicted, so we will have four SAE models, each one specialized in one particular layer.

For the experimentation, we resort to the same architecture proposed in the original work. An encoder with four blocks composed of a convolutional layer of 32 filters of 3×3, a sub-sampling of 2×2, a batch normalization, a Rectified Linear Unit (ReLU) activation, and a dropout of 0.4. On the decoder side, the blocks follow the same scheme except for the sub-sampling, which is replaced by an oversampling of the same rate. The last layer of the decoder is connected to a convolution with one 3×3 filter and a sigmoid activation to obtain the result of the prediction with values between 0 and 1. This architecture was only changed to add the masking layer after the input.

Note that each SAE was trained using the binary cross-entropy loss for up to 200 epochs with a batch size of 16, and an early stopping criterion of 20 epochs of no improvement on the validation set. Adam optimizer [34] was used with a learning rate of 0.001.

Furthermore, to favor the convergence of the model, the input images were normalized in the range $[0, 1]$. The mask was applied over this result, so the inputs can actually contain the values $\{-1\} \cup [0, 1]$. For the extraction of patches, a value of λ of 2.5% was used, since it allowed obtaining chunks with sufficient information. In addition, we also considered standard data augmentation to increase data variability by applying random rotations between -45° and 45°, zoom variations between 0.8x and 1.2x, and horizontal and vertical flips.

5. RESULTS

This section presents and discusses the results obtained with the proposed method.

First, a preliminary experiment was carried out to analyze the influence of the amount of oversampling. For this, starting from a single annotated patch, we studied the result obtained by increasing the number of randomly extracted samples around the annotated patch using the proposed technique. Fig. 2 shows the average results of this experiment in the validation set for all layers and considering both the application and non-application of data augmentation. For a small number of randomly extracted samples, the proposal achieves approximately 30% of $\mathrm{F}_1^{\mathrm{m}}$. The average result is improved as the number of samples extracted increases, reaching over 70% of $\mathrm{F}_1^{\mathrm{m}}$ for 512 samples and barely improving for the case of 1 024 samples. Additional data augmentation does not help to improve the results, only for cases of sample size equal to or less than 128. This may be because the proposed oversampling method can be considered as a data augmentation process, so that, from a given amount of sampling, there is enough variability and other techniques of data augmentation may not be necessary.

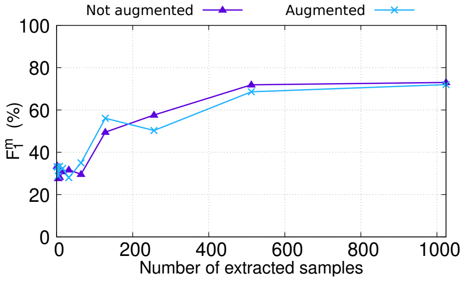


Figure 2: Preliminary experiment to study the influence of the number of samples drawn randomly from one annotated patch of 256 × 256 pixels. The result obtained in terms of $\mathrm{F}_1^{\mathrm{m}}$ (%) in the validation partition is shown, considering both the application and the non-application of additional data augmentation.

Based on these results, the sampling size is set to 512 for the following experiments. Also, since standard data augmentation seems detrimental in combination with our proposal, we decided not to use it.

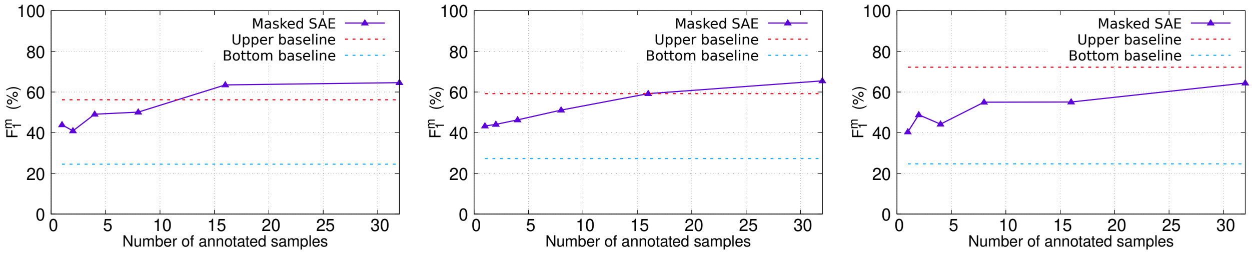
The selected configuration was evaluated using the test set, carrying out an analysis of the influence of the number of patches annotated (from 1 to 32) and the influence of these being extracted from the same page or from several (up to 4, which would generate more variability). Fig. 3 shows these results compared to two baselines: an upper one representing the state-of-the-art model [18] trained with all available information (if the entire training set was annotated) and a lower bound training this model with only one annotated patch (in both cases without applying the proposed masking layer). One initial observation is that the three case studies (with 1, 2, or 4 pages) demonstrate comparable trends. The results, as expected, show an increasing trend with the number of annotated samples, from an average $\mathrm{F}_1^{\mathrm{m}}$ of 40% when training with one annotated sample to ∼62% when using 32 annotated patches, and stabilizing (or improving less) from 16 to 32 annotated patches.

If these results are compared with the baselines, it can be seen how the proposal exceeds the lower bound by 16% when training with one annotated sample and that it equals or even improves the upper baseline in the cases with 1 and 2 pages from 16 annotated samples. Also, it is only 7% worse than the state of the art for the 4-page case but with a much lower annotated data requirement (32 samples, which represents 39% of the total information).

Layer	**Annotated samples**						**Baseline**	
Corpus	**1** **(1%)**	**2** **(2%)**	**4** **(5%)**	**8** **(10%)**	**16** **(20%)**	**32** **(39%)**	**Bt** **(1%)**	**Up** **(100%)**
staff								
EIN	10.5	39.8	64.1	62.1	83.9	78.1	0.0	87.3
SAL	72.0	75.4	75.7	75.7	74.8	87.4	0.0	90.8
MS73	11.3	13.9	17.7	12.9	92.8	94.1	0.0	91.4
CAP	66.2	75.6	75.2	79.0	79.9	82.5	0.0	47.0
Avg.	40.0	51.2	58.3	57.4	82.9	85.5	0.0	79.1
note								
EIN	19.0	16.7	20.7	0.0	20.4	26.3	0.0	77.8
SAL	35.3	3.3	21.2	4.1	38.6	50.2	0.0	4.1
MS73	0.2	3.2	6.7	7.3	7.1	7.3	0.0	2.7
CAP	66.7	69.7	73.0	77.9	81.2	82.6	0.3	8.3
Avg.	30.3	23.2	30.4	22.3	36.8	41.6	0.1	23.2
text								
EIN	22.9	15.1	17.2	67.3	31.7	37.0	11.3	11.3
SAL	67.6	15.5	46.1	32.3	71.7	73.4	0.0	78.5
MS73	6.3	9.4	26.2	16.7	15.3	14.3	0.0	13.5
CAP	3.6	0.0	15.1	37.0	45.4	16.7	3.6	12.7
Avg.	25.1	10.0	26.2	38.3	41.0	35.4	3.7	29.0
background								
EIN	93.9	93.8	93.8	93.8	93.8	93.7	93.7	93.7
SAL	93.2	93.2	93.2	93.2	97.9	99.1	93.2	98.5
MS73	40.0	36.8	46.6	49.2	87.4	96.8	96.8	96.8
CAP	93.6	93.6	93.7	93.6	93.6	93.6	93.6	93.6
Avg.	80.2	79.4	81.8	82.5	93.2	95.8	94.2	95.7

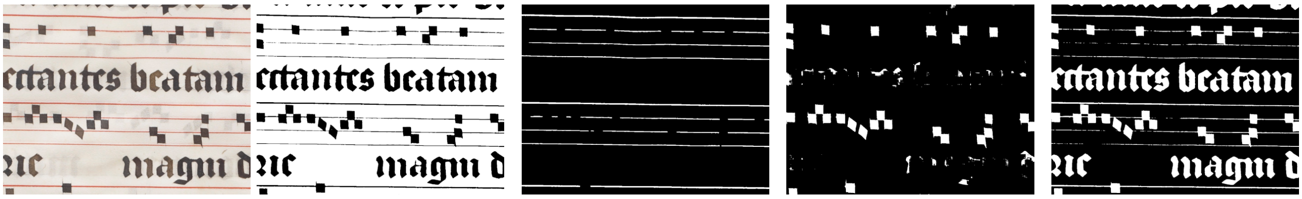
Table 2: Average results in terms of F_1 (%) for each layer considering 1 page in a few-shot evaluation. The percentage of annotated information is indicated between parentheses. **Bt** represents the bottom baseline, which is the state-of-the-art model trained with 1 annotated sample per page, and **Up** is the upper baseline, with full pages used for training. Both baselines do not apply any masking.

From these results, we now analyze in detail the case of a single page, since it represents the most extreme case as it has less variability available for the annotation. Table 2 shows a summary of the results obtained individually for each dataset and layer considered, including the baselines and the percentage of the image used in each case. As in the previous results, it is observed that the performance of our approach improves as more annotated samples are used. In this case, we can analyze how the results vary according to the layer and the corpus evaluated. In general, the proposal improves the bottom baseline, in some cases, such as `staff`, `notes`, and `text`, by a wide margin. However, note that this baseline fails to converge on most layers, except for the `background` one. In this case, on average, the proposal only improves the baseline by using 32 samples—39% of the image. This is due to the fact that for fewer annotated samples, poor overall results are obtained for the MS73 dataset. This may be because this dataset presents a greater variability of backgrounds. In



(a) 1 training page. (b) 2 training pages. (c) 4 training pages.

Figure 3: Average results in terms of F_1^m (%) with respect to the number of annotated samples (from 1 to 32) and the number of pages (from 1 to 4). Dashed lines represent baseline results for reference. The upper reference line indicates the results of the state-of-the-art model trained with fully annotated pages, while the lower reference line represents the results obtained when only one sample is annotated. Note that both baselines do not use the proposed masking method.



(a) SAL (input). (b) Background layer. (c) Staff layer. (d) Notes layer (e) Text layer.

Figure 4: Example of the results obtained in SAL for the four layers considered in this work. The method was trained with 32 samples drawn from one page. White represents the detected information for the particular layer.

fact, the rest of the layers of that corpus also obtain low-performance values when the annotated data is scarce.

Regarding the upper baseline, it can be seen how the proposal, on average, improves it in all layers, although it requires a different number of labeled samples depending on the layer. As stated before, on average, from 16 patches or 20% labeling, a better result is achieved. It is interesting that for the simplest and more homogeneous layers (such as `staff` and `background`), the upper baseline obtains a better result and it is more difficult for the proposal to overcome it, while for the more difficult ones that present greater variability (`notes` and `text`), the baseline obtains a worse result while the proposal achieves a greater margin of improvement. This may be due to the fact that the proposal performs some overfitting in the simplest cases with less variability and, therefore, requires a greater number of labeled samples to learn it.

To complement the quantitative results, Fig. 4 shows an example of prediction for SAL. As can be seen, the background and the staff layers are correctly retrieved, and some false positives can be found in the notes layer. The text layer seems the most challenging as it is not able to differentiate the ink of the text from other elements. However, the text is recovered, and the false positives could be removed by combining the predictions obtained for the other layers.

6. CONCLUSIONS

In this work, we presented a few-shot neural approach for pixel-wise layout analysis of music score images. The proposal includes a masking layer, which acts as a regularizer, that is combined with an oversampling technique to leverage the limited annotated information available. The oversampling technique extracts annotated parts of the images at different random positions at each training iteration, leaving annotated and non-annotated information in different positions of the input. This strategy forces the neural architecture to generalize the learned weights, similar to a data augmentation process but adapted to the case of few-shot and partial annotation in documents.

The proposal is evaluated on four benchmark datasets to study the influence of the amount of annotated data in the layout analysis task. We found that the number of annotated samples is key to optimizing performance, and annotating a relatively small number of them—between 16 and 32 samples, which represents using only between 20% and 39% of the total information—can achieve average results of 65.5% of F_1^m, which is very close to the result obtained by the state of the art (72%) using the entire training set annotated. It is also interesting to note that the proposal obtains similar results when labeling more pages, so it is enough to have a single page for training and perform a partial annotation of between 16 and 32 patches.

In general, the approach shows very competitive results in few-shot scenarios. Therefore, we hope this research can open doors to new avenues in this line. Reducing the amount of annotated data required for pixel-wise layout analysis is essential, and techniques such as domain adaptation and transfer learning may help to reduce human effort. We plan to investigate new ways to address this problem, including to combine domain adaptation techniques with our masking proposal and studying the feasibility of incremental and active learning.

7. ACKNOWLEDGMENT

This work was supported by the I+D+i project TED2021-132103A-I00 (DOREMI), funded by MCIN/AEI/10.13039/501100011033. This work also draws on research supported by the Social Sciences and Humanities Research Council (895-2013-1012) and the Fonds de recherche du Québec-Société et Culture (2022-SE3-303927).

8. REFERENCES

[1] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, no. 2, pp. 95–121, 2001.

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marçal, C. Guedes, and J. S. Cardoso, "Optical music recognition: State-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.

[3] J. Calvo-Zaragoza, J. H. Jr., and A. Pacha, "Understanding optical music recognition," *ACM Comput. Surv.*, vol. 53, no. 4, Jul. 2020.

[4] J. Calvo-Zaragoza, F. J. Castellanos, G. Vigliensoni, and I. Fujinaga, "Deep neural networks for document processing of music score images," *Applied Sciences*, vol. 8, no. 5, p. 654, 2018.

[5] I. Fujinaga and G. Vigliensoni, "The art of teaching computers: The SIMSSA optical music recognition workflow system," in *27th European Signal Processing Conference, EUSIPCO, A Coruña, Spain, September 2-6*. IEEE, 2019, pp. 1–5.

[6] F. J. Castellanos, A. J. Gallego, and J. Calvo-Zaragoza, "Unsupervised domain adaptation for document analysis of music score images," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 81–87.

[7] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[8] N. R. Howe, "Document binarization with automatic parameter tuning," *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 247–258, 2013.

[9] T. Pinto, A. Rebelo, G. A. Giraldi, and J. S. Cardoso, "Music score binarization based on domain knowledge," in *5th Iberian Conference on Pattern Recognition and Image Analysis, Las Palmas de Gran Canaria, Spain*, 2011, pp. 700–708.

[10] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "An MRF model for binarization of music scores with complex background," *Pattern Recognition Letters*, vol. 69, no. Supplement C, pp. 88–95, 2016.

[11] J. A. Burgoyne and I. Fujinaga, "Lyric extraction and recognition on digital images of early music sources," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009, pp. 723–728.

[12] V. B. Campos, J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Sheet music statistical layout analysis," in *15th International Conference on Frontiers in Handwriting Recognition, Shenzhen, China*, 2016, pp. 313–318.

[13] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 753–766, 2008.

[14] J. Dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. Pinto da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1134–1139, 2009.

[15] T. Géraud, "A morphological method for music score staff removal," in *International Conference on Image Processing*, 2014, pp. 2599–2603.

[16] A. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138–48, 2017.

[17] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "One-step detection of background, staff lines, and symbols in medieval music manuscripts with convolutional neural networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China*, 2017, pp. 724–730.

[18] F. J. Castellanos, J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Document analysis of music score images with selectional auto-encoders," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 256–263. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/93_Paper.pdf

[19] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[20] J.-M. Lee and D.-s. Kang, "Improved method for learning data imbalance in gender classification model using da-fsl," *Multimedia Tools and Applications*, pp. 1–19, 2021.

[21] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, "Revisiting metric learning for few-shot image classification," *Neurocomputing*, vol. 406, pp. 49–58, 2020.

[22] M. Cogswell, F. Ahmed, R. B. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," in *4th International Conference on Learning Representations, ICLR*, 2016.

[23] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[24] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2019.

[25] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning (ICML) - Deep Learning workshop*, vol. 2, 2015, pp. 1126–1135.

[27] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[28] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[29] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[30] S. Jadon, "An overview of deep learning architectures in few-shot learning domain," *arXiv preprint arXiv:2008.06365*, 2020.

[31] F. Medhat, D. Chesmore, and J. Robinson, "Masked conditional neural networks for environmental sound classification," in *Artificial Intelligence XXXIV: 37th SGAI International Conference on Artificial Intelligence, AI 2017, Cambridge, UK, December 12-14, 2017, Proceedings*. Springer, 2017, pp. 21–33.

[32] M. Suresh, A. Sinha, and R. Aneesh, "Real-time hand gesture recognition using deep learning," *International Journal of Innovations and Implementations in Engineering*, vol. 1, 2019.

[33] J. Calvo-Zaragoza, D. Rizo, and J. M. I. Quereda, "Two (note) heads are better than one: Pen-based multimodal interaction with music scores," in *Proceedings of the 17th International Society for Music InformationRetrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 509–514.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980