Pattern Recognition Letters 167 (2023) 1–8

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Few-shot symbol classification via self-supervised learning and nearest neighbor

María Alfaro-Contreras*, Antonio Ríos-Vila, Jose J. Valero-Mas, Jorge Calvo-Zaragoza

Instituto Universitario de Investigación Informática, University of Alicante, Ap. 99, 03080 Alicante, Spain

ARTICLE INFO

Article history:
Received 27 July 2022
Revised 17 December 2022
Accepted 20 January 2023
Available online 24 January 2023

Edited by Maria De Marsico

Keywords:
Symbol classification
Document image analysis
Self-Supervised learning
Few-Shot classification

ABSTRACT

The recognition of symbols within document images is one of the most relevant steps involved in the Document Analysis field. While current state-of-the-art methods based on Deep Learning are capable of adequately performing this task, they generally require a vast amount of data that has to be manually labeled. In this paper, we propose a self-supervised learning-based method that addresses this task by training a neural-based feature extractor with a set of unlabeled documents and performs the recognition task considering just a few reference samples. Experiments on different corpora comprising music, text, and symbol documents report that the proposal is capable of adequately tackling the task with high accuracy rates of up to 95% in few-shot settings. Moreover, results show that the presented strategy outperforms the base supervised learning approaches trained with the same amount of data that, in some cases, even fail to converge. This approach, hence, stands as a lightweight alternative to deal with symbol classification with few annotated data.

1. Introduction

The preservation and exploitation of cultural heritage is an essential vehicle for understanding our history and expanding new knowledge. Libraries, archives, and museums typically safeguard all the information humanity gained over the centuries. Among the oldest methods of preservation are facsimiles and handcrafted copies of historical sources. However, this approach suffers not only damage issues over use and time but also some accessibility inconveniences. For example, their study has to be performed *in situ*.

The emergence of the Internet and digital technologies produced a shift towards the digitization of those works, leading to more secure and accessible storage based on digital databases/libraries [1]. This enables not only the global accessibility of the archives but the storage of information that lies beyond documents themselves—such as metadata or content transcription—at the expense of a tedious manual transcription process.

With the advent of artificial intelligence, and specifically the rise of Machine Learning (ML) and Deep Learning (DL) strategies, alternative automated solutions appeared to ease this task. The Document Analysis field studies the comprehension and information extraction from documents by computational means [2]. This line of research has empirically demonstrated that following automated transcription approaches for document digitization reduce human-annotation effort, including that needed for revision and correction of possible errors. Although the use of DL systems offers a remarkable advantage over manual approaches, these models require to be specifically trained on the graphical domain that which the system is intended to be applied to. This means that to retrieve information from a given collection, it is mandatory to manually annotate a representative portion of data which serves as a training corpus. Indeed, the amount of training data required to obtain acceptable results tends to be vast. This is currently a bottleneck when dealing with historical documents, as there is typically few labeled data available. New avenues of research are, therefore, exploring alternative methodologies that disregard this limitation.

Self-Supervised Learning (SSL) represents one of the most recent, yet competitive, paradigms within the DL field, aimed at palliating the large amount of labeled data required by deep neural models to learn [3]. Note that, while traditional supervised learning relies on human-annotated corpora, SSL is meant to learn through pseudolabeled data–where no human annotation is involved—to then converge in one or more downstream tasks (e.g., classification) [4]. Currently, this paradigm represents an effective solution to data-lacking scenarios, providing multipurpose state-of-the-art models [5,6]. One process that can benefit from this approach is the symbol classification stage—one of high

* Corresponding author.
E-mail addresses: malfaro@dlsi.ua.es (M. Alfaro-Contreras), arios@dlsi.ua.es (A. Ríos-Vila), jjvalero@dlsi.ua.es (J.J. Valero-Mas), jcalvo@dlsi.ua.es (J. Calvo-Zaragoza).

https://doi.org/10.1016/j.patrec.2023.01.014

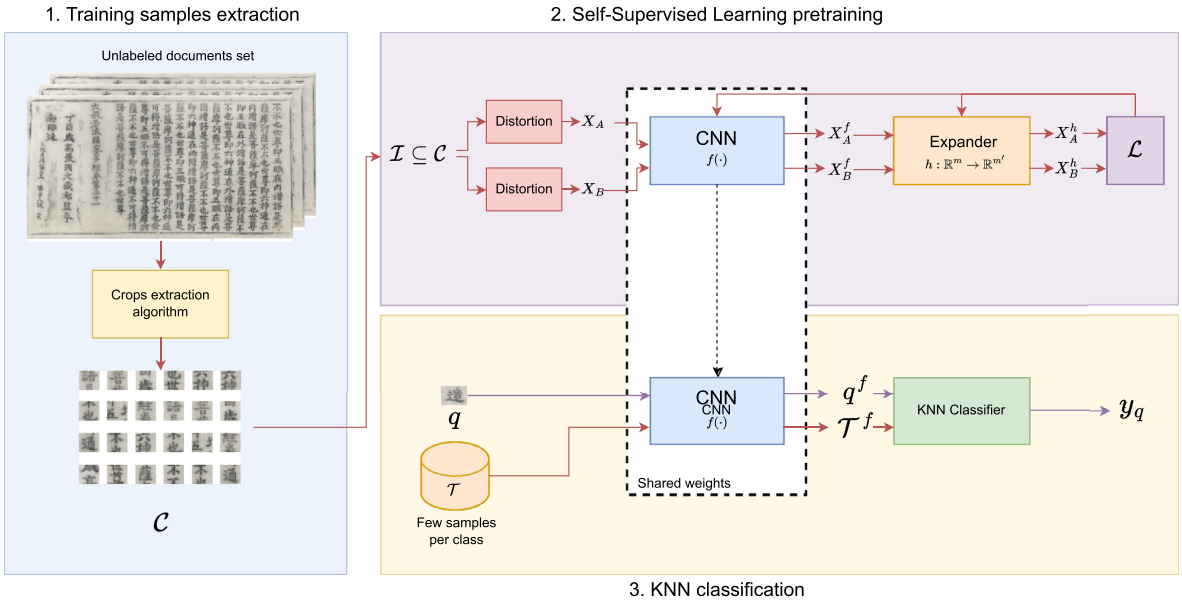


Fig. 1. General scheme of the proposed self-supervised workflow for this paper. Red arrows represent steps that are used during the workflow training—being the crop extraction and training of the neural network performed first and the *k*NN adjusting with the produced representations next—and the purple ones the inference process, where the representation of the given query is retrieved and then used in the *k*NN classifier to obtain the symbol class.

relevance—whose labeling process tends to be the most tedious and prone to errors. Although SSL stands as an interesting option, there is a challenge to tackle in this task. Symbols are typically labeled by determining not only their class but also their location in the document—i.e, marking the region they belong to with a bounding box. If SSL aims to be an interesting solution, even when addressing only symbol classification, we need the methodology to work with completely unlabeled data, where not even the position of the symbols is known.

This work proposes a symbol classification workflow that solves this scenario by sticking to the aforementioned premise. Our methodology comprises three stages: (i) data selection from completely unlabeled documents; (ii) pre-training of a deep neural model via SSL for its use as a feature extractor; and (iii) a *k*-Nearest Neighbor (*k*NN) classification [7] for labeling query samples using a remarkably reduced reference set of labeled data—namely, few-shot classification—annotated by the user.

The remainder of the paper is organized as follows: Section 2 thoroughly develops the proposed workflow. Section 3 describes the experimental setup. Section 4 presents and analyses the results, and finally, Section 5 concludes the work and discusses possible ideas for future research.

2. Methodology

This section presents the few-shot symbol classification proposal of the work, which is graphically shown in Fig. 1. In it, we train in an unsupervised manner a neural network able to generate an adequate feature representation space that can be then used by a classifier. This process comprises three stages: (i) the automatic extraction of isolated symbols from unlabeled documents; (ii) training a neural feature extractor via self-supervised learning; and (iii) a classification phase that considers the nearest neighbor rule to label symbol queries. The rest of the section thoroughly describes these stages.

2.1. Stage I: Element extraction

The first stage of the proposal aims to extract all the possible existing categories from a collection of unlabeled documents, i.e., neither class nor location annotations of the symbols within are provided. In this respect, this work proposes an algorithm to automatically extract these pieces of information by subdividing each document into a set of image patches using a sliding-window approach for then selecting those that may contain a symbol, referred to as *crops*, based on certain criteria. This proposal is now described and summarized in Algorithm 1.

Algorithm 1: Crop extraction algorithm proposed.

Input : $\mathcal{D} \leftarrow$ Set of documents.
$\omega \leftarrow$ Patch window.
$\delta_h \times \delta_w \leftarrow$ Stride factor.
Output: $\mathcal{C} \leftarrow$ Resulting set of crops.

$\mathcal{P} = \text{ExtractPatches}(\mathcal{D}, \omega, \delta_s \times \delta_h)$
$\mathcal{C} = \emptyset$ ▷ Initial empty set of crops.
for $p \in \mathcal{P}$ **do**
  $p_{bn} = \text{ConvertToGrayscale}(p)$ ▷ $p_{bn} \in \mathbb{R}^{s_h \times s_w}$
  $p_{bin} = \text{SauvolaBin}(p_{bn}, w_p)$ ▷ $p_{bin} \in \{0,1\}^{s_h \times s_w}$
  $\bar{e} = -\sum_{i \in \{0,1\}} c_i \log c_i$ where $c_i = |\{x \in P_{bin} : x = i\}|$
  if $\bar{e} > e_{\min}$ **then**
    $\mathcal{C} = \mathcal{C} \cup p$
  end if
end for

Formally, let $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$ represent a set of document images. Additionally, let ω denote an image mask of size $s_h \times s_w$ pixels, which respectively correspond to its height and width dimensions. Window ω is moved from left to right and up to down—starting from the top-right corner of the image—along all the documents in set $\mathcal{D}$ with a striding factor of $\delta_h \times \delta_w$ pixels for the height and width dimensions, respectively, retrieving a set of patches $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$, being $p_i \in \mathbb{R}^{s_h \times s_w \times c_i}$ where c_i stands for the number of channels of the image with $1 \leq i \leq |\mathcal{P}|$.

After that, a filtering process is done on set $\mathcal{P}$ to eliminate spurious and non-relevant patches for the subsequent stages by selecting a set $\mathcal{C} \subseteq \mathcal{P}$ of image portions, namely *crops*, with the following process: each patch $p \in \mathcal{P}$ is converted to gray scale—retrieving

sample $p_{bn} \in \mathbb{R}^{s_h \times s_w}$—and then binarized resorting to the Sauvola binarization method [8], hence obtaining sample $p_{bin} \in \{0,1\}^{s_h \times s_w}$; the entropy value $\bar{e}$ [9] is computed out of the p_{bin} patch and compared against a threshold $e_{\min}$—user parameter—that, if exceeded, element p is included in set $\mathcal{C}$.

2.2. Stage II: Self-Supervised neural feature extraction

The second stage of this process aims to obtain a neural-based feature extractor—specifically, a Convolutional Neural Network (CNN)—in a self-supervised manner using the set of crops $\mathcal{C}$ retrieved in the former stage. For that, we resort to the Variance-Invariance-Covariance Regularization (VICReg) method [10] due to its reported competitive overall performance in the related literature. VICReg improves previous state-of-the-art SSL methods by imposing fewer constraints on the architecture, such as avoiding the need for negative examples, which opens up the door to non-contrastive SSL. In a broad sense, this strategy allows training a neural model in a self-supervised fashion based on the so-called concepts of *variance, invariance*, and *covariance*, and whose embedded representation space meets certain conditions suitable for classification tasks.

To achieve this goal, the VICReg method initially draws an N-size batch of unlabeled image crops $I \subseteq \mathcal{C}$ that undergoes two independent image distortion processes, hence retrieving collections X_A and X_B. Note that, since these processes perform some controlled distortions in each of the images in the batch, the number of images for each collection remains the same, i.e., $|X_A| = |X_B| = N$.

After that, sets X_A and X_B are mapped to an m-dimensional space by considering a function $f(\cdot)$ given by a CNN scheme, thus obtaining collections X_A^f and X_B^f, respectively. Note that this neural model represents the actual feature extractor to be retrieved as a result of this second stage of the proposal.

Following this, an additional neural model—namely, *expander*—applies a transformation $h : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$ to sets X_A^f and X_B^f, hence producing X_A^h and X_B^h. It must be pointed out that, while not strictly necessary, this step is considered in the literature to improve the convergence of the scheme.

Finally, the expanded sets X_A^h and X_B^h are used for computing the following loss function:

$$\begin{aligned}\ell\left(X_A^h, X_B^h\right) = \lambda s\left(X_A^h, X_B^h\right) \\ + \mu\left[v(X_A^h) + v(X_B^h)\right] \\ + \phi\left[c(X_A^h) + c(X_B^h)\right]\end{aligned} \tag{1}$$

where $s(\cdot,\cdot)$ is the invariance term, which forces the network to pull these representations together in space and is computed as mean squared error (MSE), $v(\cdot)$ denotes the variance component, where a hinge loss is computed to force the network to generate information-rich vectors—by not enabling their terms to be equal—, and $c(\cdot)$ is the covariance contribution, where the embedding elements produced from a given sample are forced to be different within and prevents the *informational collapse* effect. The λ, μ, and ϕ terms of the equation represent the respective regularization multipliers for the aforementioned loss components that are experimentally tuned.

2.3. Stage III: Classification

The final stage of this proposed method performs the actual symbol classification considering the CNN-based feature extractor previously obtained. For that, we resort to the kNN classifier [7], which hypothesizes about the class of a given query attending to the labels of its closest k neighbors, based on a certain dissimilarity measure.

Formally, the initial query q and the labeled set of documents $\mathcal{T}$ are mapped to the target m-dimensional representation space as q^f and $\mathcal{T}^f$ using the CNN model obtained in the second stage of the proposal. After that, the kNN rule estimates the class y_q of query q as:

$$y_q = \text{mode}\left(\zeta\left(\underset{\forall t \in \mathcal{T}^f}{\text{argmin}}_{\,k}\left\{d\left(q^f, t\right)\right\}\right)\right) \tag{2}$$

where $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}_0^+$ represents a dissimilarity measure, $\zeta(\cdot)$ stands for the function that outputs the label of the element in the argument, and mode$(\cdot)$ denotes the mode operator.

3. Experimental setup

3.1. Corpora

Several corpora have been considered to evaluate the performance of the proposal. Note that, since the presented method aims to be a generic solution not tailored for any type of specific data, the studied datasets constitute representative examples of different domains in the Document Analysis field. The specifications of these corpora are the following.

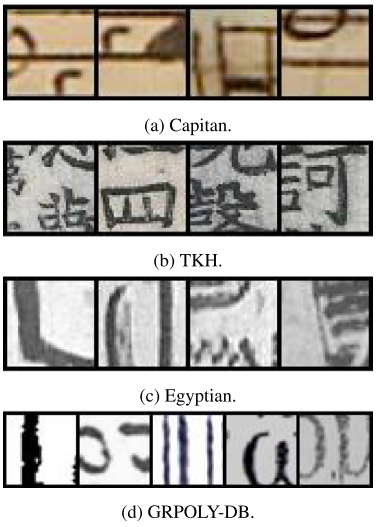
1. The *Egyptian* hieroglyph database is a collection of documents presented in Franken and van Gemert [11]. It contains ten plate photographs of ancient Egyptian hieroglyphics manually segmented and labeled, compiling approximately 172 different symbol categories.
2. The *Capitan* corpus [12] comprises 74 handwritten scores from the 17th century of a *missa* (sacred music) in mensural notation with 53 different symbols. Each page of the manuscript is provided with annotations of the individual symbols in the different music staves.
3. The *Tripitaka Koreana in Han (TKH)* dataset is a collection of Chinese historical documents created for symbol classification and detection [13]. It contains 999 document pages with 1,492 different symbol categories and their location within. This corpus was publically released by the Deep Learning and Visual Computing Lab of South China University of Technology.[1]
4. The old Greek polytonic database *GRPOLY-DB* is a collection of both machine-printed and handwritten documents presented in Gatos et al. [14]. It consists of four subsets annotated with ground-truth information at different levels. In this work, we use the "MachinePrintedA" and "MachinePrintedB" sets since their corresponding ground-truth contains segmentation at the symbol level. The total amount of data comprises 38 printed documents with 112 different symbols.

Table 1
Details of the corpora in terms of the number of samples and the cardinality of the vocabulary for each label space considered..

Corpus	Pages	Symbols	Samples
Egyptian	10	172	4,210
Capitan	74	53	17,112
TKH	999	1,492	323,498
GRPOLY-DB	38	112	168,719

Table 1 provides a summary of the characteristics of these corpora. We eventually derive two non-overlapping partitions—train

[1] https://github.com/HCIILAB/TKH_MTH_Datasets_Release



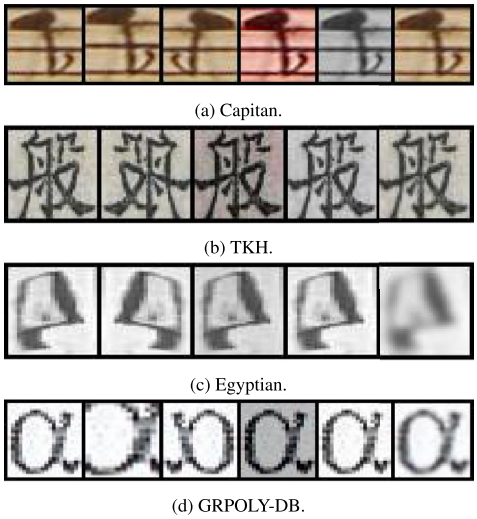
(a) Capitan.

(b) TKH.

(c) Egyptian.

(d) GRPOLY-DB.

Fig. 2. Examples of the extracted crops from Algorithm 1 in the evaluated corpora.



(a) Capitan.

(b) TKH.

(c) Egyptian.

(d) GRPOLY-DB.

Fig. 3. Example of distortions applied to a given symbol image.

and test—following a 10-iteration bootstrapping scheme for providing more robust performance figures.

3.2. Pipeline configuration

We now introduce the implementation details of the workflow proposed. For the sake of clarity, we separately detail these descriptions in their respective stages of the pipeline.

3.2.1. Stage I: Element extraction

The crop extraction proposal presented in Section 2.1 requires the specification of certain parameters to adequately address the process: the height s_h and width s_w dimensions of the image mask ω, the δ_h and δ_w displacement parameters of the sliding-window policy, and the e_{min} entropy threshold.

Based on preliminary experimentation, we eventually considered squared ω windows ($s_h = s_w = s$) of size $s = 64$ pixels for the *Capitan* and *TKH* sets and $s = 32$ pixels for the *Egyptian* and *GRPOLY-DB* corpora. Striding factor was fixed in all cases to half of the window size, i.e., $\delta_h = \delta_w = s/2$.

Regarding the entropy threshold, we set a rather restrictive value of $e_{\text{min}} = 0.8$ to avoid the selection of background or non-symbol-related data—such as staves or lyrics in the case of music documents—, which would be detrimental to the self-supervised feature extractor. Examples of the extracted information from this algorithm are shown in Fig. 2.

3.2.2. Stage II: Self-Supervised neural feature extractor

During the training stage, and as aforementioned, the proposed feature extraction method requires two neural models: (i) a CNN architecture that processes the input images to obtain an adequate embedded representation—mapping function f—; and (ii) the Expander block, which maps those features into a higher-dimensional space to ease convergence—mapping.

We consider different CNN architectures for the encoder network: a standard ResNet-34 configuration [15] and a more lightweight neural architecture that follows the topology proposed in Nuñez-Alcover et al. [16]. Other state-of-the-art models for image classification tasks such as VGG-19 [17] were evaluated in preliminary experimentation but, among all of them, the ResNet-34 backbone yielded the best results for the different considered corpora. Note that, independently of the chosen configuration, the top layer is a fully connected network that maps the representation onto the target feature space with $m = 1,600$.

In the case of the Expander block, the implementation from Bardes et al. [10] is considered. We fix $m' = 1,024$ for the Expander target dimensionality (function h) as it yielded the most promising results in preliminary experiments.

Concerning the regularization multipliers for the VICReg loss, we use $\lambda = 10$, $\mu = 1$, and $\phi = 1$ for the *Egyptian* and *TKH* sets and $\lambda = 10$, $\mu = 10$, and $\phi = 1$ for the *Capitan* and *GRPOLY-DB* corpora. Those values yielded the best convergence results in preliminary experimentation.

Finally, with regard to the image distortion processes followed to obtain sets X_A and X_B, we have resorted to a subset of those suggested in the work by Bardes et al. [10] as they are proved to provide an adequate convergence of the neural model. Fig. 3 shows an example of the different distortion processes considered in this work.

3.2.3. Stage III: Classification

Few-shot classification scenarios are defined by the N-way-L-shot approach: the training set contains N classes each with L examples [18], typically being $L < 10$ examples [19,20]. When N equals the total number of symbols of the dataset, the method is referred to as L-shot classification, being this particular scenario the one addressed in this work. We, therefore, sub-sample the reference $\mathcal{T}$ labeled set of images by randomly selecting L samples per class. For our experiments, we assess the influence of this particular parameter by considering $L \in \{1, 5, 10, 15, 20, 25, 30\}$ samples per class. It must be noted that results for values $L > 10$ are given for reference purposes as they do not constitute representative values for few-shot learning scenarios.

Regarding the kNN classifier, we set $k = 1$ to ensure the condition $k \leq L$, i.e., the number of examples of the same class is always higher than—or, at least, equal to—the number of requested neighbors. Higher values may result in $k > L$, which would remarkably hinder the performance of the scheme.

3.3. Baseline approaches

To comparatively assess the performance of the presented proposal, several non-SSL methods have been considered to establish reference results.

Table 2
Mean accuracy (%) values obtained on the classification stage in the test set for each evaluated corpus and train set $\mathcal{T}$ size. Bold figures denote the best results for each $\mathcal{T}$ labeled reference set size. The dashed line separates supervised (above) from self-supervised (below) strategies.

	Egyptian							Capitan						
	1	5	10	15	20	25	30	1	5	10	15	20	25	30
Flatten	31.8	52.7	59.2	64.7	66.4	69.5	70.6	32.6	50.4	59.9	64.4	68.0	69.7	71.9
Pre-trained ResNet34	46.9	65.0	70.0	74.0	75.7	76.8	77.8	37.0	55.7	62.6	66.3	67.8	70.3	71.0
Supervised														
Nuñez-Alcover	31.2	56.4	75.3	86.3	88.6	75.7	83.9	41.3	65.6	78.1	83.1	86.7	88.7	90.0
ResNet34	15.0	74.3	**89.3**	**93.9**	**96.4**	**97.6**	**98.1**	31.0	78.4	**90.1**	**93.3**	**96.1**	**96.7**	**97.1**
Labeled crops														
Nuñez-Alcover	**74.0**	**84.5**	88.0	89.1	90.1	90.6	90.6	65.7	82.0	86.0	88.3	89.2	90.0	90.1
ResNet34	55.2	70.4	71.8	74.5	75.4	75.9	76.9	49.2	71.0	75.4	76.7	77.9	79.3	79.2
Our approach														
Nuñez-Alcover	66.4	80.0	84.3	85.6	86.7	87.8	88.3	**67.2**	**82.0**	86.9	88.9	89.9	90.1	90.5
ResNet34	41.1	59.3	63.8	67.2	69.2	70.3	71.3	34.3	52.0	58.6	62.4	64.8	66.0	67.2

	TKH							**GRPOLY-DB**						
	1	5	10	15	20	25	30	1	5	10	15	20	25	30
Flatten	28.6	56.4	66.5	71.8	74.9	77.4	78.5	30.5	57.7	68.4	74.6	77.7	79.6	81.3
Pre-trained ResNet34	19.3	36.5	43.9	48.6	51.5	54.1	55.6	18.7	35.7	42.5	48.5	52.3	55.0	57.5
Supervised														
Nuñez-Alcover	22.5	34.1	46.2	12.5	39.6	22.5	12.5	30.0	67.6	78.4	83.5	86.1	88.2	89.1
ResNet34	27.2	83.8	93.9	96.4	**97.4**	**97.9**	**98.2**	38.0	**80.6**	**89.0**	**91.7**	**93.3**	**94.2**	**95.0**
Labeled crops														
Nuñez-Alcover	85.9	94.2	95.0	95.9	95.6	96.2	96.2	**38.6**	67.2	77.9	83.0	85.2	86.6	87.4
ResNet34	49.2	63.1	68.1	70.5	72.8	73.7	74.6	24.4	45.9	53.8	57.6	59.9	61.1	62.3
Our approach														
Nuñez-Alcover	**86.4**	**94.9**	**95.7**	**96.5**	96.4	96.7	96.7	28.6	56.8	68.7	74.3	77.0	79.2	80.6
ResNet34	32.7	53.3	60.6	64.6	66.9	68.9	70.2	16.7	31.0	37.2	40.9	43.6	45.2	46.6

The first one, namely *Flatten*, considers the individual pixels of the image as features, flattened as a one-dimensional vector. This approach serves as a reference for how the *Classification* stage considered performs without a feature extraction process, i.e., raw unprocessed images.

The second approach is the use of the ResNet-34 residual network pre-trained with the ImageNet dataset [21], which currently represents one of the state-of-the-art models for image classification tasks. This baseline, denoted as *Pre-trained ResNet34* throughout the rest of the paper, establishes a reference on how transfer learning—a common approach for few-shot learning—works for this scenario.

To provide insights on the effectiveness of traditional supervised learning in this framework, the different CNN architectures described in Section 3.2.2 with a classifier layer are used. Note that, for a fair comparison with the rest of the strategies, these methods are trained with the same few-shot set $\mathcal{T}$.

Finally, to assess the impact of using algorithmically extracted symbols from unlabeled documents as training data, we also train the self-supervised feature extractor by extracting the symbols of the given corpora by their labeled bounding boxes. This approach is referred to as *Labeled crops* in later reports.

3.4. Data-limited scenarios

To deepen the analysis of the pipeline performance presented in this article, we evaluated the proposed models on several data-constrained scenarios, where only a limited amount of crops is retrieved from the given corpora. This way, we would also draw an approximation on how many unlabeled documents are required to produce accurate classifications.

To perform this analysis, we sample the $\mathcal{C}$ set of crops by randomly selecting a subset of elements from it, retrieving reduced set $\mathcal{C}_R$. In our experiments we consider four different scenarios that differ in the number of selected elements: $\mathcal{C}_R \in \mathcal{C}$ with $|\mathcal{C}_R| = \{5k, 10k, 15k, 20k\}$. To avoid any bias when randomly sampling the $\mathcal{C}$ set of crops, these experiments are repeated 5 times to obtain a better estimate of the average performance.

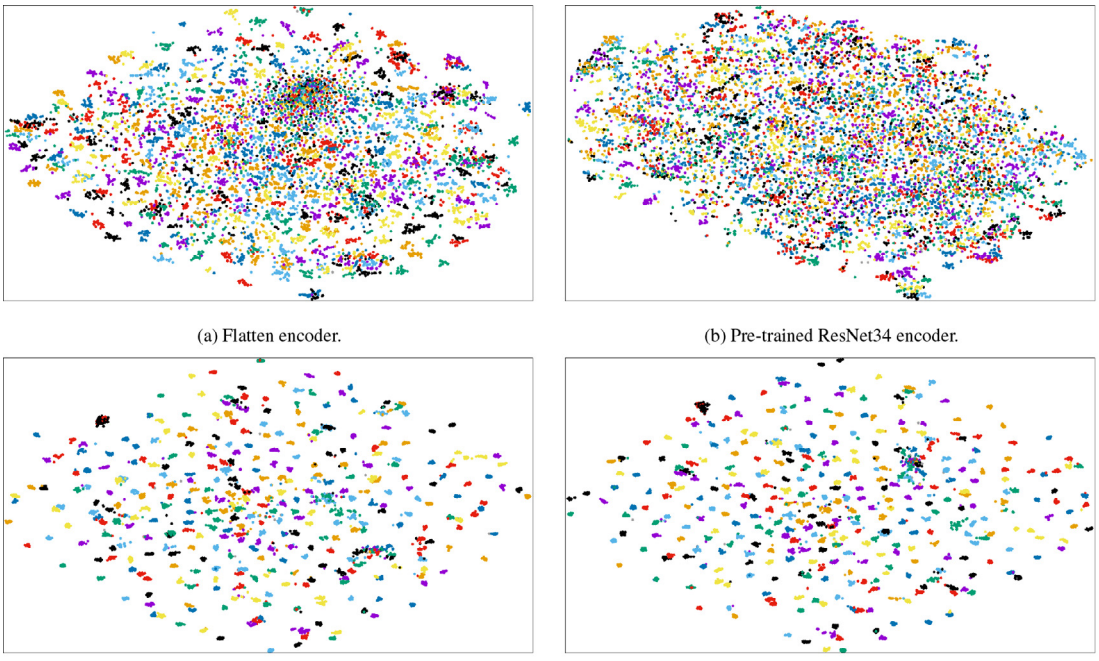
4. Results

This section discusses the results obtained with the considered experimental set-up previously posed. In this regard, Table 2 reports the average performance of both the proposed and baseline methods in terms of classification accuracy.[2]

The first idea that can be observed is that the self-supervised methods—our approach and the *Labeled crops* one—report the most competitive results when performing few-shot symbol classification, i.e., when the number of examples, L, is below 10. Specifically, it can be noticed that these methods achieve accuracy values above 60% with only one sample per class for the *Egyptian*, *Capitan*, and *TKH* datasets—obtaining at best an 86.4% score in the latter corpus—whereas the other alternatives depict performance rates below 50%. The sole exception to this is depicted by the *GRPOLY-DB* corpus for which the supervised learning approach yields the best results, being the exception the 1-shot classification scenario (although the differences can be considered marginal). The chosen values for the regularization multipliers of the VICReg loss might be inappropriate for this dataset—the performance of the downstream task is highly dependent on them as the authors of the VICReg method indicate [10].

When more examples per class are given, $L > 5$, the supervised learning approach outperforms the self-supervised one for all corpora—L must be larger than 15 for the *TKH* dataset—when the chosen configuration for the encoder network follows the

[2] The code developed in the work is publicly available for reproducible research at: https://github.com/mariaalfaroc/ssl-symbol-classification.git



(a) Flatten encoder.

(b) Pre-trained ResNet34 encoder.

(c) Self-Supervised Nuñez-Alcover pipeline with labeled crops.

(d) Self-Supervised Nuñez-Alcover pipeline trained with our algorithm.

Fig. 4. Representations of the elements in set $\mathcal{T}$ using t-SNE when considering $L = 30$ samples per class for the self-supervised methods using the Nuñez-Alcover CNN configuration as well as the *Flatten* and *Pre-trained ResNet34* supervised baseline cases for the TKH corpus. Colors in the samples denote their respective ground-truth labels.

ResNet-34 scheme. The obtained results suggest that the use of lightweight neural architectures might favor the convergence of the SSL model. Moreover, as mentioned in Section 3.2.3, this work considers an *N*-way-*L*-shot approach in which *N* equals the total number of classes. Looking at Table 1, the training set sizes would range between 530 and 14,920 samples when $L > 5$, which could be enough to favor the convergence of the supervised models. Nevertheless, the best results yielded by the two frameworks are slightly similar. Finally, it must be noted that pure transfer learning—represented by the *Pre-trained ResNet34* case—does not stand as a competitive solution in this scenario as the reported results are consistently worse than those achieved by the presented proposal.

Focusing on the self-supervised learning cases, the unlabeled-data approach obtains the best results both in the *Capitan* and *TKH* corpora, while that trained with perfectly cropped symbols yields the best performance in the *Egyptian* and *GRPOLY-DB* sets. Independently on the corpora, we observe that both self-supervised cases obtain similar overall results. That is, when there are enough training samples, having unlabeled document images from the same corpus seems to be enough to achieve adequate performance, as our crop extraction algorithm produces a set that allows the VICReg method to converge into an easy-to-classify representation space.

To further explore and support this claim, a t-distributed Stochastic Neighbor Embedding (t-SNE) [22] analysis of the representation spaces for the classification of TKH samples is presented in Fig. 4. In the case of the supervised baselines—*Flatten* and *Pre-trained ResNet34* approaches—the retrieved space is rather sparse, as no label groups are distinguished. This is not an ideal case for the nearest neighbor rule, as its best conditions are met when data can be easily separable in space, which consequently produces the results reported in Table 2. This outcome is somehow expected since the *Flatten* case directly relies on the image pixels as features while the representation space obtained when pre-training the *ResNet34* scheme with the ImageNet set does not match that of the posed symbol classification task.

In contrast to this, the representation spaces generated by the self-supervised methods using the Nuñez-Alcover CNN scheme—being this configuration the one who yielded the best overall SSL results—present sample groups that can visually be clustered, which remarkably facilitates the classification task. Moreover, the fact that these representation spaces remarkably resemble—Figs. and —supports the initial claim that the methodology presented in this paper produces similar results to the case in which all the symbols are annotated with their class and location.

Finally, Fig. 5 shows the performance results obtained when considering subsets $\mathcal{C}_R \subset \mathcal{C}$ with a limited amount of train data for the self-supervised stage using the Nuñez-Alcover CNN scheme since, as previously mentioned, it is the most competitive CNN configuration within the SSL framework. As it may be checked, limiting the $\mathcal{C}$ training data for the self-supervised learning methods noticeably affects their convergence, both in terms of accuracy and stability. In relation to the accuracy criterion, it can be observed that both the *Capitan* and *GRPOLY-DB* corpora degraded in, approximately, 3% while the Egyptian and *TKH* datasets suffer a performance drop close to 12% and 42%, respectively, when only $|\mathcal{C}_R| = 5$k samples are used. Nevertheless, when considering the largest subset of train data—$|\mathcal{C}_R| = 20$k—, the performance achieved is similar—or even better as in the case of the *TKH corpus*—to that of considering the entire crop set ($\mathcal{C}_R = \mathcal{C}$) for training the model. This fact suggests that, once provided with enough variability in the training set to retrieve a robust representation space for the nearest neighbor rule, the proposal reaches an improvement ceiling that may even degrade if more samples than necessary are provided (e.g., the TKH set).

Regarding the variability in the results, and as expected, larger $\mathcal{C}_R$ sizes increase the robustness as being the version trained with 5k samples the one that presents more dispersed performance values. Note that, while this feature varies depending on the corpus—the *TKH* variability remarkably differs from those of the *Capitan,*

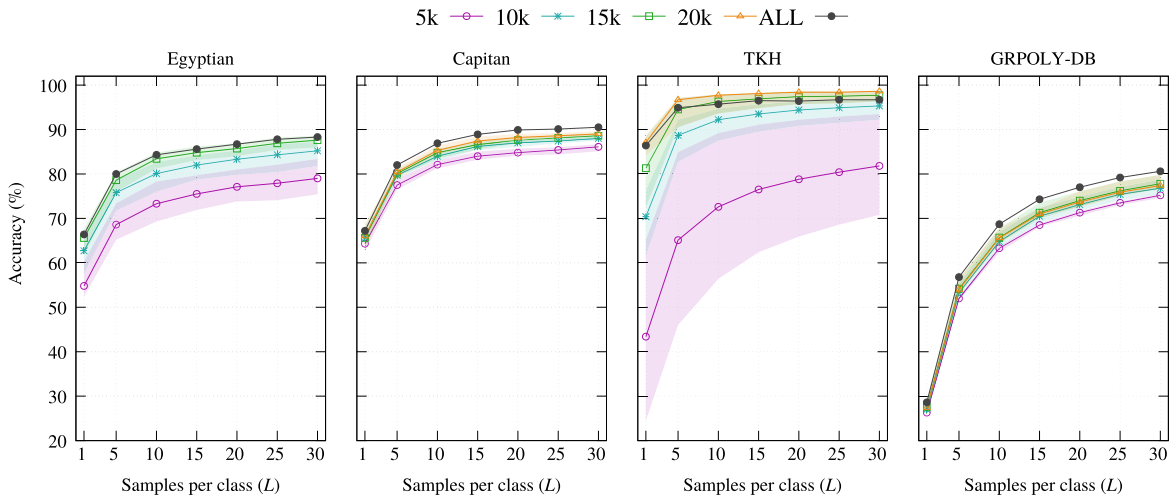


Fig. 5. Mean accuracy values (%) obtained on the data-constrained scenarios with the presented proposal using the Nuñez-Alcover CNN configuration for each of the studied corpora. Colored areas represent the dispersion in the results when considering different $\mathcal{C}_R$ random subsets.

Egyptian, and *GRPOLY-DB* corpora—it may be observed that results become remarkably stable when, at least, $|\mathcal{C}_R| = 15$k samples may be used for training the model.

5. Conclusions

This work presents a self-supervised learning method for general document recognition suited for few-shot symbol classification scenarios. This proposal comprises three different stages: (i) a first stage where symbols are automatically retrieved from unlabeled corpus; (ii) a neural-based feature extractor trained in a self-supervised manner considering the so-called Variance-Invariance-Covariance Regularization loss to generate an adequate representation space; and (iii) a k-Nearest Neighbor classifier for performing the recognition task with a considerably limited set of reference data.

The proposed method has been tested in four corpora from different domains and compared with supervised learning alternatives, ranging from simple feature extraction processes to transfer learning with state-of-the-art image-based classifiers. Additional experiments are also provided to evaluate the effectiveness of automatically retrieving the training set from unlabeled documents against extracting manually-labeled symbols.

The reported results show that the proposal outperforms the contemplated baseline strategies in terms of classification accuracy when less than 10 labelled samples per category are provided. More precisely, this strategy achieves in some corpora more than 80% of accuracy considering one single example per class in the reference set—reference strategies achieve, at most, a 30% rate. When the labelled training set is increased in size, the overall best results are attained by the supervised learning framework, although closely followed by the self-supervised ones. These differences are, however, more prominent for one of the corpora considered, suggesting that, in this case, the weights used to balance the different terms of the self-supervised loss might not be the appropriate ones.

In light of the results obtained, we consider that the self-supervised paradigm may be deemed as a suitable solution to deal with the data scarcity problem in related tasks such as layout analysis, holistic transcription, or end-to-end full-page recognition. In future work, we plan to extend this proposal to the aforementioned tasks, where novel formulations and methodologies should be proposed to enable systems to learn specific features or hierarchies in the input data that allow them to perform such tasks.

Declaration of Competing Interest

María Alfaro-Contreras reports financial support was provided by Spain Ministry of Universities. Jorge Calvo-Zaragoza reports financial support was provided by Spain Ministry of Science and Innovation. Antonio Ríos-Vila reports financial support was provided by Government of Valencia Ministry of Innovation Universities Science and Society. Jose J. Valero-Mas reports financial support was provided by Government of Valencia Ministry of Innovation Universities Science and Society.

Data availability

Data will be made available on request.

Acknowledgments

This paper is part of the project I+D+i PID2020-118447RA-I00 (MultiScore), funded by MCIN/AEI/10.13039/501100011033. The first author is supported by grant FPU19/04957 from the Spanish Ministerio de Universidades. The second and third authors are respectively supported by grants ACIF/2021/356 and APOSTD/2020/256 from “Programa I+D+i de la Generalitat Valenciana”.

References

[1] E. Duval, M. van Berchum, A. Jentzsch, G.A.P. Chico, A. Drakos, Musicology of early music with europeana tools and services, in: M. Müller, F. Wiering (Eds.), Proceedings of the 16th International Society for Music Information Retrieval Conference, 2015, pp. 632–638.

[2] J.-Y. Ramel, N. Vincent, Semantic and interaction: when document image analysis meets computer vision and machine learning, in: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20–25, 2019, IEEE, 2019, pp. 1187–1192.

[3] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 4037–4058.

[4] K. Ohri, M. Kumar, Review on self-supervised image recognition using deep neural networks, Knowl. Based Syst. 224 (2021) 107090.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[6] H. Bao, L. Dong, F. Wei, Beit: BERT pre-training of image transformers, in: 10th International Conference on Learning Representations, Apr 2022, Virtual, France, 2022.

[7] R.O. Duda, P.E. Hart, D.G. Stork, Pattern classification, 2nd, Wiley, 2001.
[8] J.J. Sauvola, T. Seppänen, S. Haapakoski, M. Pietikäinen, Adaptive document binarization, in: 4th International Conference Document Analysis and Recognition (ICDAR '97), 2-Volume Set, August 18–20, 1997, Ulm, Germany, Proceedings, IEEE Computer Society, 1997, pp. 147–152.
[9] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[10] A. Bardes, J. Ponce, Y. LeCun, VICReg: variance-invariance-covariance regularization for Self-Supervised Learning, in: 10th International Conference on Learning Representations, 2022.
[11] M. Franken, J.C. van Gemert, Automatic egyptian hieroglyph recognition by retrieving images as texts, in: Proceedings of the 21st ACM international conference on Multimedia, 2013, pp. 765–768.
[12] J. Calvo-Zaragoza, A.H. Toselli, E. Vidal, Handwritten music recognition for mensural notation with convolutional recurrent neural networks, Pattern Recognit. Lett. 128 (2019) 115–121.
[13] H. Yang, L. Jin, W. Huang, Z. Yang, S. Lai, J. Sun, Dense and tight detection of chinese characters in historical documents: datasets and a recognition guided detector, IEEE Access (2018).
[14] B. Gatos, N. Stamatopoulos, G. Louloudis, G. Sfikas, G. Retsinas, V. Papavassiliou, F. Sunistira, V. Katsouros, GRPOLY-DB: an old Greek polytonic document image database, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 646–650.
[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[16] A. Nuñez-Alcover, P.J. Ponce de León, J. Calvo-Zaragoza, Glyph and position classification of music symbols in early music manuscripts, in: A. Morales, J. Fiérrez, J.S. Sánchez, B. Ribeiro (Eds.), Pattern Recognition and Image Analysis - 9th Iberian Conference, IbPRIA 2019, Madrid, Spain, July 1–4, 2019, Proceedings, Part II, Lecture Notes in Computer Science, volume 11868, Springer, 2019, pp. 159–168.
[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[18] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples: a survey on few-Shot learning, ACM Comput. Surv. 53 (3) (2020).
[19] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, 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.
[20] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.
[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
[22] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).