Pattern Recognition Letters 160 (2022) 43–49

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Few shots are all you need: A progressive learning approach for low resource handwritten text recognition

Mohamed Ali Souibgui[a,*], Alicia Fornés[a], Yousri Kessentini[b,c], Beáta Megyesi[d]

[a] Computer Vision Center, Computer Science Department, Universitat Autònoma de Barcelona, Spain
[b] Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit,Sfax 3021 Sfax, Tunisia
[c] SM@RTS : Laboratory of Signals, SysteMs, Artificial Intelligence and Networks, Tunisia
[d] Department of Linguistics and Philology, Uppsala University, Sweden

ARTICLE INFO

Article history:
Received 8 July 2021
Revised 9 April 2022
Accepted 4 June 2022
Available online 7 June 2022

Edited by Jiwen Lu

Keywords:
Handwritten text recognition
Few-shot learning
Unsupervised progressive learning
Ciphered manuscripts

ABSTRACT

Handwritten text recognition in low resource scenarios, such as manuscripts with rare alphabets, is a challenging problem. In this paper, we propose a few-shot learning-based handwriting recognition approach that significantly reduces the human annotation process, by requiring only a few images of each alphabet symbols. The method consists of detecting all the symbols of a given alphabet in a textline image and decoding the obtained similarity scores to the final sequence of transcribed symbols. Our model is first pretrained on synthetic line images generated from an alphabet, which could differ from the alphabet of the target domain. A second training step is then applied to reduce the gap between the source and the target data. Since this retraining would require annotation of thousands of handwritten symbols together with their bounding boxes, we propose to avoid such human effort through an unsupervised progressive learning approach that automatically assigns pseudo-labels to the unlabeled data. The evaluation on different datasets shows that our model can lead to competitive results with a significant reduction in human effort. The code will be publicly available in the following repository: https://github.com/dali92002/HTRbyMatching

1. Introduction

Training data-hungry models based on deep learning in low resource scenarios is challenging due to the scarcity of labeled data. This is particularly the case of modern Handwritten Text Recognition (HTR) systems when applied to manuscripts with rare scripts or unknown alphabets. For example, ancient civilizations used their specific alphabets that are no longer used (e.g. cuneiform, Egyptian hieroglyphs) and historical ciphers (used in diplomatic and intelligence reports, secret societies, or private letters) often invented fanciful cipher alphabets to hide the content of the message [1].

Recognizing and extracting information from these special sources are important to the understanding of our cultural heritage, since it helps to shed new light on and (re-)interpret our history [2]. However, a manual transcription is impractical due to the amount of manuscripts, and automatic recognition is difficult due to the lack of labeled training data. Moreover, the problem becomes even harder in the case of ciphers because when the alphabet is invented, no dictionaries or language models are available to help in the training process.

Contrary to deep learning models, human beings are able to learn new concepts from one or a few samples only. Recent research has been conducted to imitate and simulate this ability. One of these recent approaches is called few-shot learning, requiring only a limited number of examples with supervised information [3]. In our previous work [4], we explored whether few-shot learning could be adapted to the recognition of various symbol sets in encrypted hand-written manuscripts.

Usually, HTR models must be trained on the particular alphabet to be recognized, and whenever the alphabet changes, the system must be retrained from scratch with samples from the new script. To avoid the cumbersome process of re-training, we treated the recognition as a symbol detection task: by providing one or a few samples of each symbol type in the alphabet, the system could locate the symbols in the manuscript. The model was generic and could be used for multiple scripts, while requiring only a small sample of labeled data on each new symbol type. The first experimental results obtained a good performance on encrypted

* Corresponding author.
E-mail address: msouibgui@cvc.uab.cat (M.A. Souibgui).

https://doi.org/10.1016/j.patrec.2022.06.003

manuscripts compared to the typical methods, while reducing the amount of labeled data for fine-tuning.

Nevertheless, the required labeled data in our few-shot model still implies a significant human effort: labeling a few pages with various types of symbols for fine tuning include manual transcription of thousands of symbols together with their corresponding bounding boxes. To alleviate this, we aim to minimize the time-consuming manual labeling effort by proposing an unsupervised learning approach that can automatically and progressively label the data by assigning *pseudo – labels* from the unlabeled handwritten text lines. Our method requires only a few shot of the desired alphabet: to perform the pseudo-labeling, the user crops a few samples – preferably 5 – of each symbol type thereby avoiding the annotation of text lines and the annotation of the bounding boxes. This means that the pseudo-labeled data is automatically obtained to fine-tune our model, with zero manual effort.

The main contributions of our work are: (*i*) We propose a few-shot learning model for transcribing hand-written manuscripts in low resource scenarios with minimal human effort. Our model only requires few, ideally five samples of each new symbol type, instead of annotating the entire text lines with the symbols and their bounding boxes. (*ii*) We propose an unsupervised, segmentation-free method to progressively obtain pseudo-labeled data, which can be applied to cursive texts with touching symbols. (*iii*) We propose a generic recognition and pseudo-labeling model that can be applied across different scripts. (*iv*) We demonstrate the effectiveness of our approach through extensive experimentation on different datasets with various alphabets, reaching a performance similar to the one obtained with manually labeled data.

2. Related work

2.1. Low resource manuscript recognition

A manuscript is considered low resource when it contains rare symbols or unusual symbol sets. Thus, collecting a training set for this manuscript is difficult (especially a labeled one). The research on the transcription of enciphered manuscripts is quite recent. In [5], an MultiDimensional Long Short-Term Memory (MDLSTM) [6] approach was proposed. The performance was satisfactory, but at the cost of the time-consuming manual data labeling. The method also required new, manual transcription for each new cipher. Instead, some unsupervised methods were introduced [7,8] to avoid the costly human effort. Those approaches were segmenting the enciphered documents into isolated symbols then clustering them. The main disadvantage of the clustering method turned is the segmentation of symbols, because is was often inaccurate, provoking transcription errors. Similarly, researchers have opted for learning-free symbol spotting approaches [9,10] for the transcription of ancient manuscripts (e.g. Egyptian hieroglyphs, cuneiform, or runes).

In summary, supervised methods obtain good performance but they require large amounts of labeled data, while unsupervised or learning-free methods can be applied when labeled data is not available but they lead to lower performance. Thus, to maintain high accuracy while reducing the human effort of manual labeling, few-shot learning is a promising alternative to use for handwritten text recognition [4]. A similar approach based on character matching was proposed in [11], although the experiments were mostly carried out on synthetic data, instead of on real historical or cursive manuscripts.

2.2. Handwritten text pseudo-Labeling

Pseudo-labeling models aim to take advantage from unlabeled data when training, which makes it a possible solution for low resource manuscript. In semi-supervised learning [12,13], a few labeled data is used to start the process. For instance, in the label propagation approach based on distances [14,15], labels are assigned from the unlabeled data (called pseudo-labels) to be used to reinforce the training. Similarly, in [16], the training started with some true labels that are gradually increased by pseudo labels. In [17] a shared backbone extracted features from the labeled, pseudo-labeled and unlabeled data at each iteration. Then, from the feature space, the reliable labels were estimated according to the distance with the true labels while the non trusted labels were pushed away with an exclusive loss. Besides, a pseudo-labeling curriculum approach for domain adaptation used a density-based clustering algorithm in [18]. The idea was to annotate data with the same label set, but taken from a different domain.

In HTR, this strategy was hardly applied mainly due to the difficulties in character segmentation, since touching characters are common in cursive texts. In [19], labels were guessed at word level using keyword spotting. A confidence score was used to assign new labels to the retrieved words and enlarge the dataset. Furthermore, a text to image alignment was proposed in [20] following the mentioned strategy.

3. Proposed approach

In this section we describe our approach to handwritten text recognition by few-shot learning. First, our model is trained on synthetic data. We create text line images using various Omniglot symbol alphabets [21]. Then the model is fine-tuned using the pseudo-labeling approach with the specific alphabet of the target domain, in our case the hand-written manuscript. The involved steps are described in detail below.

3.1. Few-shot manuscript matching

As explained in Section 2, few-shot modeling for object detection has shown to be suitable for recognizing the manuscripts in low resource scenarios. In few-shot modeling, if the size of the alphabet is N, and we provide k examples from each symbol alphabet (named *shots* (or supports)), the task is considered as an N-way k-shot detection problem. In such setting, the model can be trained on certain alphabets with sufficient amount of labeled data, and later, it can be tested on new alphabets (classes) with a few labeled examples only.

Our few-shot learning model, illustrated in Fig. 1, is segmentation free and works at line level. As input, it takes the text line image with an associated alphabet in the form of isolated symbol images. In this step, from one to five samples of each alphabet symbol should be given. The two inputs (the line image as a query and a symbol image as a support) are propagated in a shared backbone to derive two feature maps. Those are then used in the Region Proposal Network (RPN) with an attention mechanism to output proposals. The attention mechanism performs the depth-wise cross correlation between the support and query feature maps. As illustrated in Fig. 2, this is done by performing a multiple average pooling to the support feature map to obtain a shape of $1 \times 1 \times Channels$ then multiplying it over depth with the query feature map. After the RPN stage, the Region of Interest (ROI) pooling is applied to the RPN proposals and the support image to provide two feature maps having the same size. These feature maps are representing the support image as well as the query regions that are candidates to match the support image. Those are combined together and passed to the final stage where the bounding boxes are produced with the class 1 (similar to the support) or 0 (different from the support symbol). For each labeled bounding box, a confidence score between 0 and 1 is predicted according to

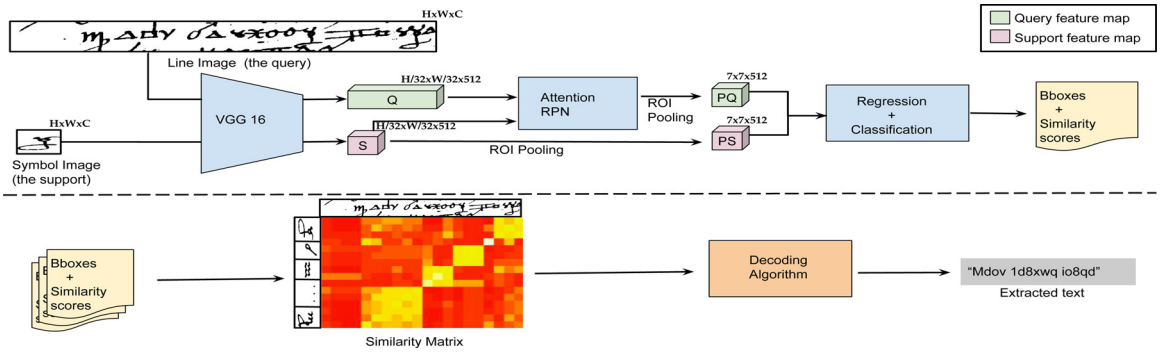

Fig. 1. Our few-shot approach for handwriting recognition. Examples of each symbol in the alphabet are used as supports. Up: Detection of a support symbol in a handwritten line. Down: Construction of the similarity matrix from the predicted bounding boxes and its decoding to obtain the final text.

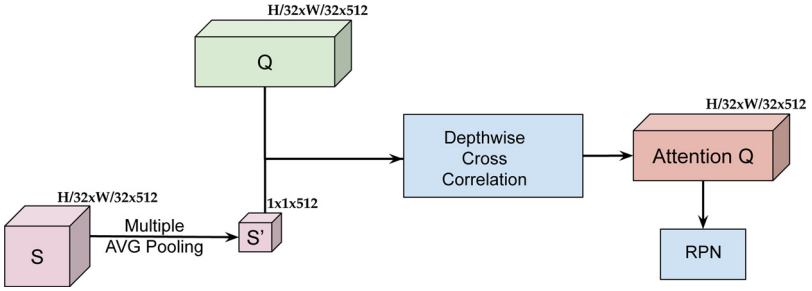

Fig. 2. An illustration of the attention RPN: the support feature map is average pooled until obtaining a tensor with shape of $1 \times 1 \times 512$. The obtained tensor is multiplied over depth with the Query feature map to obtain the attention Q, which is passed to the RPN for region proposing.

the similarity degree with the support image. We repeat this process for all supports (i.e. all the alphabet symbols) and take only the bounding boxes with a high confidence score (higher than a given threshold) to construct a similarity matrix between the symbol alphabet and the line image regions. This matrix serves as the input to the decoding algorithm, which provides the final transcription.

3.2. Similarity matrix decoding

The decoding stage, detailed in Algorithm 1, takes the similarity matrix, traverses the columns from left to right, and decides for each pixel column the final transcribed symbol class among the candidate symbols. Concretely, for each time step, it chooses the symbol having the maximum similarity score. To minimize errors, a symbol is only transcribed if its bounding box is not overlapped by another symbol with a higher similarity value for a certain number of successive pixels. In our case, we used 15 pixels as a threshold. Despite its simplicity, this decoding method is effective for transcribing sequences of symbols. It can be considered also as a modified version of the Connectionist Temporal Classification (CTC) algorithm [22].

As mentioned before, our few-shot model is first trained on the Omniglot dataset: we synthetically construct lines to learn the matching in different alphabets. Then, at testing time, it can be used to recognize new symbols, requiring only a support set composed of a few examples of each new symbol class. However, in our previous work [4], experiments showed that the predictions can be significantly improved when we fine-tuned the model using some real text lines, due to the domain difference between the synthetic Omniglot symbols and the real historical symbols.

3.3. Progressive pseudo-labeling

Our proposed progressive data pseudo-labeling strategy consists in two stages described below.

Algorithm 1 Similarity Matrix Decoding.

Input:
  M ▷ Similarity matrix
  rep_thresh ▷ Repetition threshold
Output: $CharList$ ▷ Characters sequence
  $last\_max \leftarrow [-1, 0]$ ▷ $[index, score]$
  $repetition \leftarrow 0$
  $maximums \leftarrow MaxInd(M)$ ▷ maximum index and score for each column
  $CanAdd \leftarrow False$

  for $maxi$ `in` $maximums$ **do**

    if $maxi \neq last\_max$ **then**
      $repetitions \leftarrow 0$
      $CanAdd \leftarrow True$
    else
      if $repetitions > rep\_thresh$ `and` $CanAdd$ **then**
        $CharList \leftarrow CharList \cup maxi[index]$
        $CanAdd \leftarrow False$
      else
        $repetitions \leftarrow repetitions + 1$
      end if
    end if
  end for

3.3.1. Synthetic data generation

Our few-shot model needs to be fine-tuned using data from the target domain (often with an unseen alphabet) to reduce the gap between the source and target domains. But since we aim to minimize the user effort, we restrain the demands on a support set of few samples from each new symbol alphabet. Hence, the user must only select up to 5 samples per symbol, called shots. From those shots, we automatically generate synthetic lines by randomly

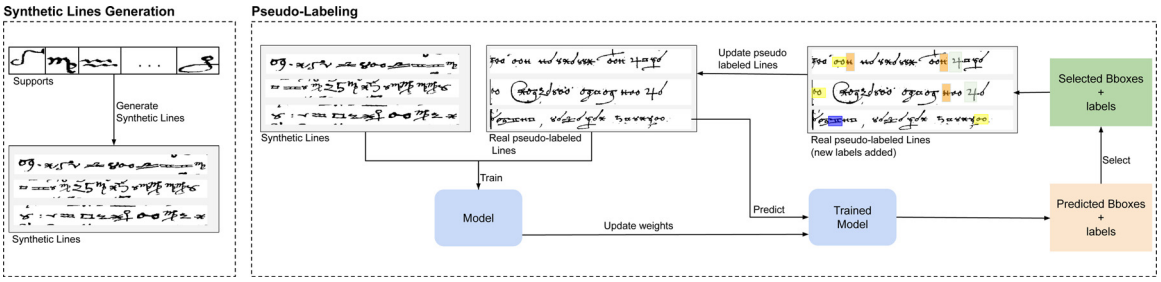

Fig. 3. Our pseudo-labeling approach: In the beginning, synthetic lines are generated using the support set. Then, the pseudo-labeling phase starts. At starting, there is no pseudo-labeled data, so only synthetic lines will be used for retraining the model. Then, the model predicts symbols from the real unlabeled lines with the same script. The symbols with highest confidence score, namely pseudo-labels, are labeled and added with their predicted bounding boxes. Next, the model is retrained again using the synthetic lines and the pseudo-labeled symbols from real lines. The process is repeated until the full dataset is annotated.

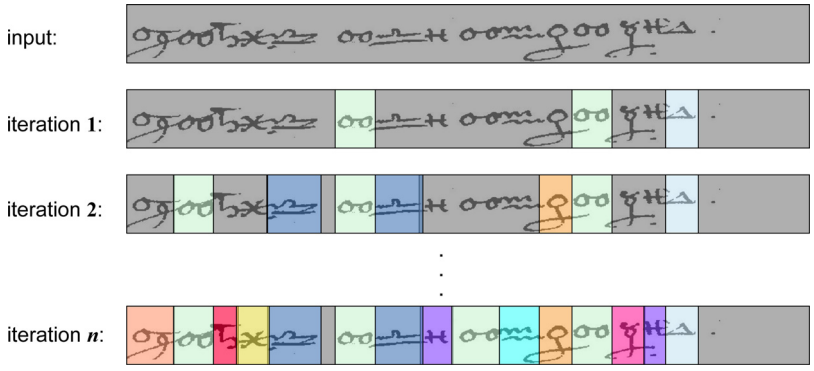

Fig. 4. An example of pseudo-labeling of a line image. The background is colored in grey, while the predicted label classes at each time are shown in colors. Each symbol class is shown with a different color (best viewed in color).

concatenating them in a line image. We tried to make those synthetic lines as realistic as possible. To do so, the space between characters was chosen randomly between 0 and 30 pixels. Also, before concatenation we rotate each character randomly between -5 and 5 degrees. Moreover, we add some artifacts to the upper part and lower part of the line to simulate a realistic segmentation of a handwritten line. Those created lines compose our starting labeled set, since our model was only pre-trained on a different data domain, i.e. the synthetic Omniglot lines. This technique significantly improves the model prediction for unseen alphabets or scripts.

3.3.2. Pseudo-labeling process

After retraining our model with synthetic lines, we begin by annotating the unlabeled data. The process is illustrated in Fig. 3. Of course, at the beginning, the pseudo labeled set is empty (as no labels are available), so only the synthetic lines can be used for training. Then, we pass the real text lines through our model to get the predictions, which include the bounding boxes of the regions that are similar to the input alphabet images along with the assigned similarity score. Since a higher score means more credible label, we choose the top scored predictions as pseudo labels at this iteration. We experimentally found that the best option is to choose, at each iteration, the 20% of the training data size as the number of the new pseudo-labels. The obtained pseudo-labeled set will be joined to the synthetic set for the next training iteration. This process is repeated until the whole unlabeled set (i.e. all text lines) is annotated. In the case where it is not possible to add new pseudo-labels with credible confidence score, we set a threshold of 0.4 as the minimum confidence score for assigning pseudo-labels. In fact, whenever the score is below this threshold, it is better not to label the symbol. Note that we label the handwritten lines without the need of segmenting them into isolated symbols. In this way, the remaining unlabeled symbols in the different lines at each iteration are considered as background during the next training. Fig. 4 shows an example of a handwritten line during the pseudo labeling process. At the beginning, the whole image is considered as a background. Then, the symbols with higher confidence score are labeled in the first iteration, while the hardest ones will be labeled in the next iterations.

4. Experiments

4.1. Datasets

For low resource handwritten text, we chose two historical encrypted manuscripts and a manuscript with an old, not longer used alphabet, the Codex Runicus. Our choice of running experiments on encrypted manuscripts is motivated by the fact that ciphers contain a large variety of more or less fancy symbols instead of or in addition to using common alphabets and/or digits. In this work we chose two encrypted manuscripts, namely the Borg and the Copiale ciphers, both containing a large variety of symbols. The Borg[1]. cipher is a 408 pages long manuscript, originating from the 17th century. The entire manuscript is encoded with the exception of the first and last two pages, and some headings in Latin. The cipher consists of 34 different symbols, comprising from graphic signs to Latin letters and some diacritics. The Copiale cipher is a 105 page long encrypted manuscript from the mid 18th century. The cipher consists of 100 different symbols from Latin and Greek letters to digits along with a large number of graphic signs. The cipher has been transcribed and deciphered [23] and is freely available in high resolution images[2]. The Codex Runicus[3] is a historical manuscript, the oldest preserved Nordic provincial law written on 100 parchment folios of 202 pages. Its symbol set consists

[1] https://cl.lingfil.uu.se/~bea/borg/
[2] https://cl.lingfil.uu.se/~bea/copiale/
[3] https://www.e-pages.dk/ku/579/

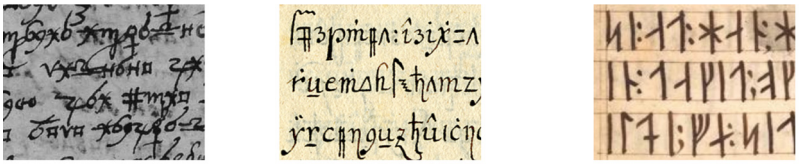

Fig. 5. Examples of the three manuscripts with low resource annotated data.

Table 1
Obtained Results on the different datasets. FT: Fine Tuning. Om: Omniglot. SD: Synthetic Data. RLD: Real Labeled Data. PLD: Pseudo Labeled Data. ULD: UnLabeled Data.

Dataset	Method	User Effort	Training → FT	SER
Borg	Unsupervised [8]	None	ULD	0.57
	Unsupervised [8]	Manual Segmentation	ULD	0.22
	Unsupervised [7]	Clusters Processing	ULD	0.54
	MDLSTM [5]	Manual Labeling	RLD	0.55
	Few-shot [4]	Manual Labeling	Om → RLD	0.21
	Few-shot [4]	5 shots	Om → NONE	0.53
	Ours	5 shots	Om → SD + PLD	0.24
Copiale	Unsupervised [8]	None	ULD	0.44
	Unsupervised [8]	Manual Segmentation	ULD	0.37
	Unsupervised [7]	Clusters Processing	ULD	0.20
	MDLSTM [5]	Manual Labeling	RLD	0.07
	Few-shot [4]	Manual Labeling	Om → RLD	0.11
	Few-shot [4]	5 shots	Om → NONE	0.39
	Ours	5 shots	Om → SD + PLD	0.15
Codex	Unsupervised [7]	Clusters Processing	ULD	0.06
Runicus	MDLSTM [5]	Manual Labeling	RLD	0.26
	Few-shot [4]	Manual Labeling	Om → RLD	0.05
	Few-shot [4]	5 shots	Om → NONE	0.40
	Ours	5 shots	Om → SD + PLD	0.09

,

Table 2
Required time (in minutes) for manually annotating the training lines.

Dataset	# Lines	# Symbols	# Classes	Time
Borg	117	1913	24	≈ 245
Copiale	176	7197	78	≈ 450
Runicus	56	1583	25	≈ 206

of runes where each rune corresponds to a letter of the Latin alphabet. Fig. 5 shows examples of the two ciphers and the codex Runicus. As it can be seen, the Borg symbols are connected not only horizontally but also oftentimes vertically with many touching symbols making its recognition challenging. In the Copiale cipher, on the other hand, the symbols are clearly segmented but the size of the alphabet is large, which make it a good challenge to test our approach on it. Similar to the Copiale cipher, the codex Runicus consists of clearly segmented symbols and a rare alphabet making it also a good case of low resource handwriting recognition. In our experiments, we exclude the symbols with low frequencies (that occur once or twice) in all manuscripts. We use 24 symbols from the Borg cipher, 78 symbols from the Copiale cipher, and 25 symbols from the Codex Runicus. Table 2 shows more information about our used datasets.

4.2. Experimental setup and metrics

To carry out the experiments, we first trained our proposed few-shot handwriting recognition model using lines created from the Omniglot dataset only. Then, we retrained the model using synthetic lines created from the given 5 symbols (shots) as described above. This data is called Synthetic Data (SD). Afterwards, we start predicting the labels and obtaining the Pseudo Labeled Data (PSD) by using the approach detailed in Section 3.3. We finally fine-tune the model with the pseudo-labeled data and compare its performance to the models that uses Real Labeled Data (RLD) for training.

The model performance is measured by the Symbol Error Rate (SER) metric. It is the same as the Character Error Rate used in HTR. Formally, $SER = \frac{S+D+I}{N}$, where S is the number of substitutions, D of deletions, I of insertions and N is the ground-truth's length. Not surprisingly, the lower the value, the better performance.

We compare our approach with our previous few-shot model [4], the unsupervised [7,8] and supervised [5] approaches for encrypted manuscript recognition.

5. Results

Table 1 shows the obtained results. The Borg manuscript is considered to be a hard case because of the overlapping symbols, which makes predicting correct bounding boxes challenging. Also, the writing style is variable. As it can be seen, using a few-shot method with real labels leads to a SER of 0.21, being considered as the upper bound. But, this result is costly, since a user must manually annotate 1913 symbols, including their labels and bounding boxes. We also notice that the supervised MDLSTM with a larger training set, annotated at line level (but without any bounding boxes required), obtains a moderate result, probably due to the connected hand-writing. We notice that the unsupervised methods are only useful when the segmentation of lines into isolated symbols is accurate, which is a costly and difficult task. Our few-shot model, trained on Omniglot only and tested on Borg, leads also to a poor result (a SER of 0.53) due to the big difference between the training and test domains. However, when using the pseudo-labeled data provided by our approach, we obtained an acceptable

result of 0.24 SER, with a high gain in user effort because we only require 5 examples of each symbol, avoiding a time-consuming manual annotation.

The Copiale manuscript contains easy to segment symbols but with a larger alphabet size. As it can be seen from Table 1, the MDLSTM performs better on this dataset because of the larger labeled training lines and a unique handwriting style. However, our model achieves a competitive result by using less data than MDLSTM. Anyway, annotating these lines is costly, so a better choice is to automatically produce pseudo-labels. By using our pseudo-labeling process, we achieve a competitive performance, compared to the manually labeled data (a SER of 0.15 versus 0.11).

Finally, we test our method on the Runicus manuscript as an example of ancient document with a rare alphabet. This manuscript can be considered easier than ciphers because the symbol segmentation is easy and the alphabet size is moderate. Thus, an unsupervised clustering method can be also appropriate. Using our method with real labeled data, we obtain results that are better than without any fine tuning, with a SER of 0.05 and 0.40, respectively. When we compare the quality of our produced-pseudo labels against the manually created ones, we observe that, by using pseudo-labeling, we achieve a competitive result of 0.09 SER. This demonstrates the suitability of our method, because the performance is close to the one obtained with manual labels while significantly reducing the annotation effort.

We can conclude that our proposed pseudo-labeling method achieves good results when recognizing low resource handwritten texts, with an important decrease in the user effort for data annotation. The analysis of the human effort is detailed next.

5.1. Annotation time consumption

Manually annotating data is a time consuming task and it should be taken into account when using HTR models. Thus, in this section, we measure the time needed to label the three datasets to illustrate the manual labeling effort. As shown in Table 2, the more lines and the bigger the alphabet size, the more time is required to label the symbols with their bounding boxes. For reference, we measured the required time for providing the shots for our method and compared it with the manual annotation time. We found that locating and cropping 5 examples of each symbol in the alphabet takes approximately 40 s. Thus the user needed to spend 16 min for Borg, 17 min for Runicus and 52 min for Copiale for providing the shots for all symbols in the manuscripts for our approach.

We can conclude that automatically providing pseudo-labels significantly minimizes manual effort with a minimal loss in recognition performance compared to the manual annotation.

5.2. Pseudo-labeling performance analysis

Our proposed method progressively labels the dataset: we start by labeling easy symbols and progressively label the complicated ones. As a consequence, the accuracy of correctly labeling bounding boxes decreases as we select new pseudo labels at each iteration. We evaluate the quality of our pseudo-labeling approach on the three datasets by comparing the predicted bounding boxes and their corresponding pseudo-labels to the manually annotated ones. A predicted bounding box is defined as a correct detection if it has a minimum overlap (i.e. Intersection over Union: IoU) of 0.7 with the ground-truth box. We find that the more difficult the dataset is in terms of segmentation, alphabet size and similarity between symbols, the more the performance of our pseudo-labeling approach decreases and the more iterations in the labeling process are needed. For example, labeling accuracy for the Borg cipher was 74% after obtaining all the labels. In Copiale, where symbols are easy to segment, the labeling accuracy reached 85%. In Codex Runicus, we obtained the highest pseudo-labeling accuracy of 94% because the symbol segmentation is easier than Borg and the number of classes is lower than in Copiale.

Table 3
The symbol error rate when using different thresholds while pseudo-labeling the data. Thres.: Threshold.

Thres.	SER	Thres.	SER	Thres.	SER	Thres.	SER
0.8	0.27	0.6	0.25	0.4	0.24	0.2	0.25

Table 4
Comparative results with self-supervised learning approaches in the semi-supervised scenario.

Method	Labeled lines	SER
MAE [24]	20%	0.99
	30%	0.99
	50%	0.95
UP-DETR [25]	20%	0.71
	30%	0.70
	50%	0.66
Ours	20%	0.25
	30%	0.24
	50%	0.20

During our experiments, we found that it is better to continue the pseudo-labeling process despite the decreasing performance. The reason is that, although we might add some wrong labels, in general, the incorporation of difficult examples benefits the training and even a bounding box with a wrong label is still helping in the segmentation part. Moreover, the experiments show that there is a small difference in performance between the manually annotated labels and our automatically produced ones, which encourages us to further improve our labeling process.

5.3. Selecting threshold for pseudo-labeling

In our experiments we set a threshold of 0.4 before adding a character into the labeled set. This threshold is chosen after testing other values and finding that 0.4 is the optimal one. We show the results of the conducted experience in Table 3, where we tested different thresholds to select the pseudo-labels. The experiments were carried out on the Borg dataset.

5.4. Is semi-supervised learning worth to use?

So far, we opt to use an unsupervised approach that starts from a few shots of the desired alphabet. However, the choice of starting by labeling some real lines (text and bounding boxes) and pseudo-labeling the rest is also a possible solution. We test this strategy on the Borg dataset as presented in Table 4. For comparison, we use two recent self supervised learning methods: masked autoencoders (MAE) [24] and UP-DETR [25]. Given that these methods were proposed for image classification and object detection, we adapt them to text recognition by adding a transformer decoder [26] in MAE and using our decoding algorithm in UP-DETR. As pre-training, MAE uses a set of Latin handwritten images because of the very few available Borg lines, while the UP-DETR is trained on the 117 unlabeled Borg lines. Then the fine-tuning is done using 20, 30 and 50% of labeled Borg lines for both methods. The obtained results show that the more labeled lines, the better the performance. Despite the very few data that were used (the full training set is 117 lines), our method clearly outperforms the data-hungry MAE and UP-DETR, showing that they are not suitable for such low resource scenarios.

It is noteworthy that, in our method, if we start with more manually labeled lines, the amount of unlabeled lines to pseudo

label is reduced, so the training time decreases. Overall, we can conclude that starting from only a few shots is a better solution with regards to the reduced manual effort, since the SER is slightly affected (we obtain 0.24 as SER using our unsupervised pseudo-labeling).

6. Conclusion

We have presented a novel pseudo-labeling transcription method for manuscripts with rare alphabets or few labeled data. Our method can significantly reduce the human labor of annotating historical manuscripts, while maintaining the recognition performance. The performed experiments on the enciphered and historical manuscripts confirmed the usefulness of our approach, with a significant reduction in user effort and a minimal loss in recognition performance.

Our few-shot model with pseudo-labeling is a significant extension of our previous work [4]. In fact, its simplicity makes it even applicable on top of other methods, like [11]. Also, for widely-used alphabets (like latin) but with few labeled data, pseudo-labels can be predicted to annotate the data and train usual fully supervised HTRs, which may lead to better results than the few-shot ones.

In future, we aim to enhance the quality of the provided labels to keep reducing the need of manual intervention. Also, we plan to extend our approach to cover more low resource datasets including other unknown scripts.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Swedish Research Council, grant 2018-06074, DECRYPT–Decryption of Historical Manuscripts, the Spanish project RTI2018-095645-B-C21 and the CERCA Program / Generalitat de Catalunya. We thank the MTA Cloud (https://cloud.mta.hu/), which we used to run experiments.

References

[1] B. Megyesi, N. Blomqvist, E. Pettersson, The decode database: Collection of historical ciphers and keys, in: Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019, June 23–26 2019, Mons, Belgium, 2019, pp. 69–78.
[2] B. Megyesi, B. Esslinger, A. Fornés, N. Kopal, B. Láng, G. Lasry, K.d. Leeuw, E. Pettersson, A. Wacker, M. Waldispühl, Decryption of historical manuscripts: the decrypt project, Cryptologia 44 (6) (2020) 545–559.
[3] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples: a survey on few-shot learning, ACM Comput. Surv. (CSUR) 53 (3) (2020) 1–34.
[4] M.A. Souibgui, A. Fornés, Y. Kessentini, C. Tudor, A few-shot learning approach for historical ciphered manuscript recognition, in: Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 5413–5420.
[5] A. Fornés, B. Megyesi, J. Mas, Transcription of encoded manuscripts with image processing techniques, in: Proceedings of the Digital Humanities Montreal, Canada, August 8–11, 2017.
[6] A. Graves, Offline Arabic handwriting recognition with multidimensional recurrent neural networks, in: Guide to OCR for Arabic Scripts, Springer, 2012, pp. 297–313.
[7] A. Baro, J. Chen, A. FornĘs, B. Megyesi, Towards a generic unsupervised method for transcription of encoded manuscripts, in: Proceedings of the International Conference on Digital Access to Textual Cultural Heritage (DATECH), 2019.
[8] X. Yin, N. Aldarrab, B. Megyesi, K. Knight, Decipherment of historical manuscript images, in: Proceedings of the ICDAR, IEEE, 2019, pp. 78–85.
[9] L. Rothacker, D. Fisseler, G.G. Müller, F. Weichert, G.A. Fink, Retrieving cuneiform structures in a segmentation-free word spotting framework, in: Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing, 2015, pp. 129–136.
[10] B. Bogacz, N. Howe, H. Mara, Segmentation free spotting of cuneiform using part structured models, in: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2016, pp. 301–306.
[11] C. Zhang, A. Gupta, A. Zisserman, Adaptive text recognition through visual matching, in: Proceedings of the ECCV, 2020, pp. 51–67.
[12] X.J. Zhu, Semi-supervised Learning Literature Survey, University of Wisconsin–Madison Department of Computer Sciences, 2005.
[13] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, T. Raiko, Semi-supervised learning with ladder networks, arXiv preprint arXiv:1507.02672(2015).
[14] M.S.T. Jaakkola, M. Szummer, Partially labeled classification with markov random walks, Adv. Neural Inf. Process. Syst. (NIPS) 14 (2002) 945–952.
[15] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural networks: Tricks of the Trade, Springer, 2012, pp. 639–655.
[16] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Proceedings of the Workshop on Challenges in Representation Learning, ICML, 3, 2013.
[17] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, Y. Yang, Progressive learning for person re-identification with one example, IEEE Trans. Image Process. 28 (6) (2019) 2872–2881.
[18] J. Choi, M. Jeong, T. Kim, C. Kim, Pseudo-labeling curriculum for unsupervised domain adaptation, in: Proceedings of the British Machine Vision Conference (BMVC), Springer, 2019.
[19] V. Frinken, M. Baumgartner, A. Fischer, H. Bunke, Semi-supervised learning for cursive handwriting recognition using keyword spotting, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition, IEEE, 2012, pp. 49–54.
[20] G. Leifert, R. Labahn, J.A. Sánchez, Two semi-supervised training approaches for automated text recognition, in: Proceedings of the 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2020, pp. 145–150.
[21] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.
[22] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 369–376.
[23] K. Knight, B. Megyesi, C. Schaefer, The Copiale Cipher, in: Proceedings of the Invited Talk at ACL Workshop on Building and Using Comparable Corpora (BUCC), Association for Computational Linguistics, 2011.
[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, arXiv preprint arXiv:2111.06377(2021).
[25] Z. Dai, B. Cai, Y. Lin, J. Chen, Up-detr: unsupervised pre-training for object detection with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1601–1610.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).