In Codice Ratio: OCR of Handwritten Latin Documents using Deep Convolutional Networks

Donatella Firmani[1], Paolo Merialdo[1], Elena Nieddu[1], and Simone Scardapane[2]

[1] Roma Tre University
donatella.firmani@uniroma3.it,merialdo@dia.uniroma3.it,ema.nieddu@gmail.com
[2] Sapienza University
simone.scardapane@uniroma1.it

Abstract. Automatic transcription of historical handwritten documents is a challenging research problem, requiring in general expensive transcriptions from expert paleographers. *In Codice Ratio* is designed to be an end-to-end architecture requiring instead limited labeling effort, whose aim is the automatic transcription of a portion of the Vatican Secret Archives (one of the largest historical libraries in the world). In this paper, we describe in particular the design of our OCR component for Latin characters. To this end, we first annotated a large corpus of Latin characters with a custom crowdsourcing platform. Leveraging over recent progresses in deep learning, we designed and trained a deep convolutional network achieving an overall accuracy of 96% over the entire dataset, which is one of the highest results reported in the literature so far. Our training data are publicly available.

Keywords: deep convolutional neural networks, handwritten text recognition, optical character recognition, medieval documents

1 Introduction

Historical documents are an essential source of knowledge concerning past cultures and societies [10]. Until recently, the main bottleneck was the availability of large collections of historical documents in digital form. Today, many historical archives have begun instead a full digitalization of their assets, including the Bibliothèque Nationale de France[1] and the Vatican Apostolic Library.[2] Due to the cost (and time) required for manual transcription of these documents, and the sheer size of the collections, the challenge has become the design of fully automatic solutions for their transcription in computer-readable form. While impressive results have been achieved for printed historical documents [15], successfully transcribing handwritten documents remains a challenging task due to

[1] http://gallica.bnf.fr/
[2] http://www.digitavaticana.org/

a variety of reasons, including irregularities in writing, ligatures and abbreviations, errors in transcription, and so forth (see the discussion in Section 2).

In Codice Ratio is an interdisciplinary project involving Humanities and Engineering departments from Roma Tre University, as well as the Vatican Secret Archives, aiming at the complete transcription of the Vatican Registers, a corpus of more than 18000 pages contained as part of the Vatican Secret Archives, with minimal labeling effort. The Vatican Secret Archives is one of the largest historical libraries in the world, containing more than 85 linear kilometres of shelving. Interestingly, 'secret' does not stand for confidential, but rather denotes them as private property of the Pope. The corpus is comprised of official correspondence of the Roman Curia produced in the 13th century, including letters, opinions on legal questions, addressed from and to kings and sovereigns, as well as to many political and religious institutions throughout Europe. Never having been transcribed in the past, these documents are of unprecedented historical relevance, and could shed light to that crucial historical period. A preliminary description of the system appeared in [1].

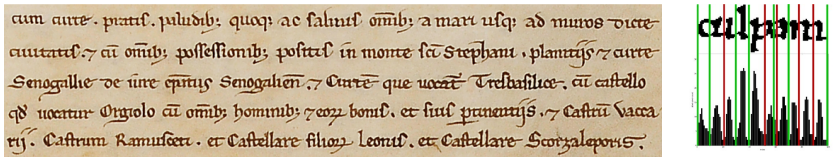
Our contribution. In this paper, we describe the design of a novel component for optical character recognition (OCR) of the Latin characters extracted from the text. Building a corpus for this task is extremely challenging due to the complexity of segmenting the characters and reading ancient fonts [5]. For this project, we implemented a custom crowdsourcing platform, employing more than a hundred high-school students to manually label the dataset. After a data augmentation process, the result was the creation of an inexpensive, high-quality dataset of 23000 characters. Following recent progresses in deep learning [8], we designed a deep convolutional neural network (CNN) for the classification step. In the last years, deep CNNs have become the *de facto* standard for complex OCR problems [2, 3]. Our trained deep CNN achieves an overall accuracy of 96% on an independent test set, which is one of highest results obtained in the literature so far. The aim of this paper is to show the effectiveness of the classification step, and the evaluation of the pipeline in [1] is out of our current scope.

Structure of the paper. The rest of the paper is structured as follows. After discussing related projects in Section 2, we detail the construction of our annotated dataset in Section 3, and the design (and training) of the CNN in Section 4. We experimentally evaluate the network in Section 5, before discussing future works in Section 6.

2 Related Work

Due to the many challenges involved in a fully automatic transcription of historical handwritten documents, many researchers in the last years have focused on solving easier sub-problems, most notably keywords spotting [11]. However, as more and more libraries and archives worldwide digitize their collections, great effort is being put into the creation of full-fledged transcription systems [4].

One of the largest effort to this end was the EU-funded tranScriptorium project [12], which resulted, among others, in the transcription of a relatively



(a) (b)

Fig. 1: (a) Sample text from the manuscript *Liber septimus regestorum domini Honorii pope III*, in the Popes' Registers of the Vatican Secret Archive. (b) Proposed segmentation cut-points for the word 'culpam'. We use green for actual character boundaries, and red otherwise.

large corpus of Dutch handwritten documents from the 15th century. Several competitions have been organized on the datasets released from the tranScriptorium project [13]. State-of-the-art algorithms from these challenges generally work by a segmentation-free approach, where it is not necessary to individually segment each character.[3] While this removes one of the hardest steps in the process, it is necessary to have full-text transcriptions for the training corpus, in turn requiring expensive labeling procedures with expert paleographers on the period under consideration. To overcome this limitation and reduce the training costs, *In Codice Ratio* focuses on a character-level classification, allowing us to collect a large corpus of annotated data using a cheap crowdsourcing procedure.

3 Dataset Collection

The dataset is collected from high-resolution (300 dpi, 2136×2697 pixels) scans of 30 pages coming from register 12 of Pope Honorii III. All pages are in the so-called Caroline minuscule script, which spread in Western Europe during Charlemagne's reign and became a standard under the Holy Roman Empire. Compared to similar fonts, writings in the Caroline minuscule are relatively regular and have fewer ligatures. A sample text is shown in Fig. 1a.

All pages are pre-processed according to the workflow in [1], by first removing the background, splitting the text into lines, and then extracting tentative character' segmentations as shown in Fig. 1b. Each tentative character is then fed to the OCR system, built on top of a deep CNN, described in the next section. A further sub-system based on a Hidden Markov Model is then in charge of selecting the most probable word transcription starting from all the possible segmentations of the word. In this paper we focus on the design of the OCR system, and we refer to [1] for a more accurate description of the first and third steps.

Character classes. We take into account minuscule characters of the latin alphabet, yielding initially 19 classes (a, b, c, d, e, f, g, h, i, l, m, n, o, p, q, r, s, t, u) plus one special non-character class $\otimes$. Since our dataset includes multiple

[3] Segmenting and recognizing a character are two heavily interdependent processes: this is known as Sayre's paradox [14].

(a) d1 (b) d2 (c) s1 (d) s2 (e) s3

Fig. 2: Different shapes of the characters “d” and “s”.

versions of characters “d” and “s”, we split class d into two classes (d1 and d2), and class s into three (s1, s2 and s3). The different character shapes and the corresponding labels are shown in Fig. 2. We have total 23 classes, including 22 character classes and the special non-character class $\otimes$.

Crowdsourcing. To collect annotations on the segmentations of the manuscript words, a custom crowdsourcing platform was developed. We enrolled 120 high-school students in the city of Rome, that did the labeling as a part of a work-related learning program. The task to perform was simple: having positive and negative examples for a given character, each student was required to select any matching images from a grid appearing on the platform. In Fig. 3, we show a screenshot of a task.

Fig. 3: Sample screen of our platform.

Each task consists of 40 images, arranged in a grid, each with its own check-box. Every time the check-box is marked, the image receives a vote. Image labels correspond to the most voted characters, among those with at most 3 votes.[4] If there is no such character, the image is labelled with a special non-character class, denoting a wrong segmentation.

Characters with less than 1K examples were augmented to match the required quantity and balance the training set. The augmentation process involves slight random rotation, zooming, shearing and shifting, both vertical and horizontal. Before training, all image values are normalized in the range [0,1]. The final dataset comprises 23K examples evenly split between 23 classes, and is available online[5].

4 Network Architecture

Our deep CNN takes as input 56×56 single-channel images, which are binarized before training. The input is then propagated through 8 adaptable layers, whose design is inspired to similar networks having achieved state-of-the-art results in

[4] In our experiments, we did not observe any tie.

[5] `http://www.dia.uniroma3.it/db/icr/`.

modern OCR recently [8]. First, we apply a convolutional layer having 42 filters with size 5×5 and stride 1. Secondly, the output of the convolutional layer is fed to a rectified linear (ReLU) nonlinearity applied element-wise:

$$g(s) = \max\{0, s\} \ . \tag{1}$$

The output of the ReLU is down-sampled using a max-pooling operation with stride 2×2 to reduce the number of adaptable parameters. The previous three operations (convolution, nonlinearity, and max-pooling) are repeated another two times, using 28 filters for the convolutional layer instead of 42. The output of the last convolutional layer is then flattened and fed through a fully connected layer with 100 neurons and ReLU nonlinearities, and a final output layer with a softmax activation function to output a probability distribution over the 23 classes.

In order to prevent overfitting, we apply 50% dropout during training [8] before each of the nonlinearities. We minimize a regularized cross-entropy loss given by:

$$J(\boldsymbol{w}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} -\hat{y}_{i,k} \log(y_{i,k}) + \lambda \|\boldsymbol{w}\|^2 , \tag{2}$$

where N is the number of examples in the training dataset, $K = 23$ is the number of classes, $y_{i,k}$ is the correct output of the kth class over the ith input, $\hat{y}_{i,k}$ is the predicted output of the network, $\boldsymbol{w}$ is the vector of adaptable parameters of the network, and $\lambda > 0$ is a regularization factor. The regularization factor is selected as $\lambda = 0.001$ by doing a grid-search over different values in an exponential interval and computing the accuracy on a held-out validation set of 2500 examples from the original training set. This validation set is also used to select a stopping point for the optimization procedure. We minimize (2) using the Adam algorithm [9] on randomly sampled mini-batches of 128 elements until the validated accuracy stops improving (200 epochs), using default hyperparameters as in [9]. The final network is then tested on a further independent test set of another 2300 examples.

5 Experimental Results

Overall accuracy reached 96%, while average precision, recall and F1-measure for each class are reported in Table 5 (support is always 100). The confusion matrix is shown in Fig. 4. Some typical errors are the following.

- Characters "f" and "s1" are easily confused, due to their similar shapes. Specifically, $\approx 8\%$ of "s1" are labelled as "f", and 14% of "f" as "s1".
- Images not containing any character are sometimes mis-classified as actual characters, mainly as "m" Specifically, $\approx 10\%$ of "not-character" are labelled as "m", and $\approx 15\%$ of "not-character" are labelled as some other character.

For comparison purposes, we report that a simple logistic regression classifier on the same dataset achieves average 80% precision and 79% recall.

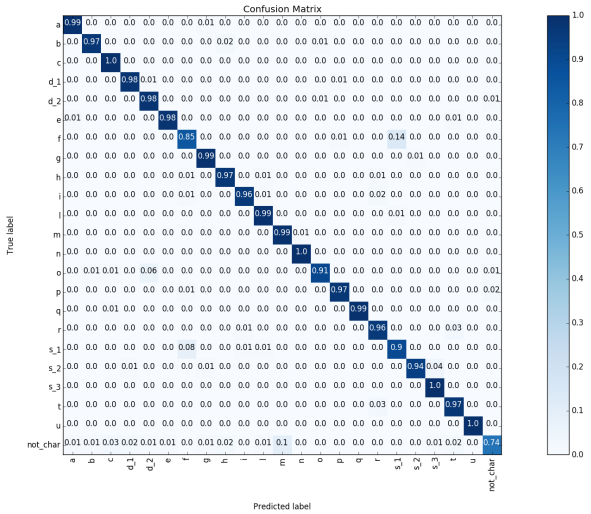


Fig. 4: Confusion matrix for the test set.

Convolution visualization. We show in Fig. 6a the effect of the filters learned by of our network at the first level. Specifically, we show the result of convolution with first layer filters on a sample input image after the activation function (blues are positive values). Visually inspecting activation output is indeed useful for debugging purposes. In the figure, the effect of edge and lighting detection filters is clearly visible.

Gradient Ascent Given the filters learned by our network, we now perform *gradient ascent* over the input image (initially random) and maximize the output of each filter, separately. This is a common step to visualize what the network has learnt to recognize [16]. Intuitively, we generate synthetic images that maximize the activation of the filters of each layer, including the output layer. In deep CNNs, the first layers usually detect simple features, and the features become more complex and abstract as the layers go deeper. The result of this experiment for a sample of our filters is shown in Fig. 6b. The figure suggests that the first layer of our network is in charge of detecting edges, while the second layer exhibits more complex, geometrical patterns. Finally, the third and deepest layer, seems to detect whole character strokes.

	Prec.	Rec.	F1
a	0.98	0.99	0.99
b	0.98	0.97	0.97
c	0.95	1.00	0.98
d1	0.97	0.98	0.98
d2	0.92	0.98	0.95
e	0.99	0.98	0.98
f	0.89	0.85	0.87
g	0.97	0.99	0.98
h	0.96	0.97	0.97
i	0.98	0.96	0.97
l	0.96	0.99	0.98
m	0.91	0.99	0.95
n	0.99	1.00	1.00
o	0.98	0.91	0.94
p	0.98	0.97	0.97
q	1.00	0.99	0.99
r	0.94	0.96	0.95
s1	0.86	0.90	0.88
s2	0.99	0.94	0.96
s3	0.95	1.00	0.98
t	0.94	0.97	0.96
u	1.00	1.00	1.00
$\otimes$	0.95	0.74	0.83
avg	0.96	0.96	0.96

Fig. 5: Per-class results.

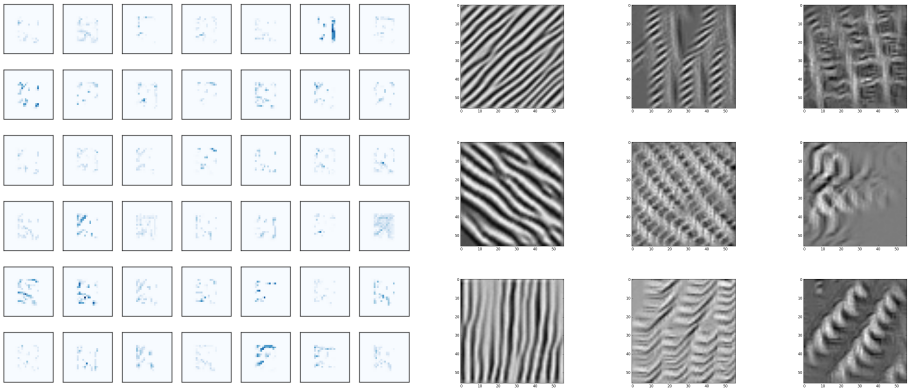


Fig. 6: (a) Input image "q" convoluted with kernels from the first convolutional layer. (b) Some examples of gradient ascent generated images over the filters of each convolutional layer.

6 Conclusions

In this paper, we have described the collection of a large corpus of annotated Latin characters, and the design of a novel deep convolutional network for the classification step. The described system is a key component in the *In Codice Ratio* project, whose aim is to fully transcribe a large corpus of documents contained in the Vatican Secret Archives [1]. Some preliminary results with the entire system have shown that the framework is able to reach around 80% of word-error rate on the pages under consideration. Thorough evaluation of the entire system (including the segmentation step) is ongoing work.

Future work will require the design of a fully differentiable system to substitute the currently hand-tuned segmentation step. Recently, indeed, some authors have proposed the use of recurrent networks to process the entire text sequentially [6, 7]. While these methods still require the annotation of the entire text, annotations can be noisy, and obtained results are generally higher than related systems based on hidden Markov models.

Acknowledgments

We thank Debora Benedetto, Elena Bernardi and Riccardo Cecere for their help with the pre-processing steps and the crowd-sourcing application. Finally, we are indebted to all the teacher and students of Liceo Keplero and Liceo Montale who joined the work-related learning program, and did all the labeling effort.

References

1. S. Ammirati, D. Firmani, M. Maiorino, P. Merialdo, E. Nieddu, and A. Rossi. In codice ratio: Scalable transcription of historical handwritten documents. In *25th Italian Symposium on Advanced Database Systems (SEBD)*, 2017. To Appear.
2. D. Cireşan and U. Meier. Multi-column deep neural networks for offline handwritten chinese character classification. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2015.
3. D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010.
4. A. Fischer. *Handwriting recognition in historical documents*. PhD thesis, Universitat Bers, 2012.
5. A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz. Ground truth creation for handwriting recognition in historical documents. In *9th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 3–10. ACM, 2010.
6. A. Fischer, M. Wuthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz. Automatic transcription of handwritten medieval documents. In *15th IEEE International Conference on Virtual Systems and Multimedia (VSMM)*, pages 137–142. IEEE, 2009.
7. V. Frinken, A. Fischer, H. Bunke, and R. Manmatha. Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents. In *2010 International Conference On Frontiers in Handwriting Recognition (ICFHR)*, pages 352–357. IEEE, 2010.
8. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
9. D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations (ICLR)*, 2015.
10. J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, , J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.
11. M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós. Efficient segmentation-free keyword spotting in historical document collections. *Pattern Recognition*, 48(2):545–555, 2015.
12. J. A. Sánchez, V. Bosch, V. Romero, K. Depuydt, and J. de Does. Handwritten text recognition for historical documents in the transcriptorium project. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 111–117. ACM, 2014.
13. J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal. Icfhr2014 competition on handwritten text recognition on transcriptorium datasets (HTRtS). In *2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 785–790. IEEE, 2014.
14. K. M. Sayre. Machine recognition of handwritten words: A project report. *Pattern Recognition*, 5(3):213–228, 1973.
15. U. Springmann, D. Najock, H. Morgenroth, H. Schmid, A. Gotscharek, and F. Fink. OCR of historical printings of latin texts: problems, prospects, progress. In *ACM First International Conference on Digital Access to Textual Cultural Heritage (DATeCH)*, pages 71–75. ACM, 2014.
16. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, Cham, 2014.