READ
RECOGNITION & ENRICHMENT OF ARCHIVAL DOCUMENTS

D8.15
Large Scale Demonstrators - Venice Time Machine

Meta-learning model

Sofia Ares Oliveira, Frederic Kaplan
EPFL

Distribution: http://read.transkribus.eu/

READ
H2020 Project 674943

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943

Project ref no.	H2020 674943
Project acronym	READ
Project full title	Recognition and Enrichment of Archival Documents
Instrument	H2020-EINFRA-2015-1
Thematic priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date/duration	01 January 2016 / 42 Months

Distribution	Public
Contract. date of delivery	31.12.2018
Actual date of delivery	10.12.2018
Date of last update	December 10, 2018
Deliverable number	D8.15
Deliverable title	Large Scale Demonstrators - Venice Time Machine
Type	report
Status & version	reviewed by :
Contributing WP(s)	WP8
Responsible beneficiary	EPFL
Other contributors	
Internal reviewers	Günter Mühlberger, CVL, NAF
Author(s)	Sofia Ares Oliveira, Frederic Kaplan
EC project officer	Martin Majek
Keywords	Large Scale Demonstrators, Segmentation, Information Retrieval, Cadastral computing

Contents

1 Executive summary

The Venice Time Machine (VTM) project aims at building a multidimensional model of Venice and its evolution covering a period of more than 1000 years. The State Archives of Venice possesses an estimated 80 km of shelves that are filled with administrative documents, from birth registrations, death certificates and tax statements, all the way to maps and urban planning designs. These archives are currently being digitized, transcribed and indexed, setting the base of the largest database ever created on Venetian documents.
The Venice Time Machine project wants to give the archives a new, virtual existence on the Web through Cloud access and online tools. It aims to reanimate Venice's past life from them by re-creating social networks and family trees, and visualizing urban development and design.
As one of the four large scale demonstrators of the READ project, the objective is to provide an environment to test and use the technologies developed within the READ consortium on large real-world data and to develop new solutions to deal with large historical archive indexing and retrieval.
This report documents the progress on two systems presented in the previous year, namely the the transcription and the segmentation systems, and on their application to cadastral computing:

- Transcription system
- Document segmentation tool (with Benoit Seguin)
- Information extraction on cadastral sources
- Prototype of a spatiotemporal search engine

2 Transcription using convolutional recurrent neural networks (CRNN)

Following last year's development in the transcription system we extended the evaluation and comparison of the performances of the convolutional recurrent neural network [1] and also expanded its usage to new corpora.

2.1 Architecture

The transcription system is based on the combination of convolutional and recurrent neural networks as described in [2] for handwritten text (Fig. 1). One one hand, convolutional neural networks (CNN) capture hierarchical spatial information, with the first layers capturing low level features and later ones capturing high level features. On the other hand, recurrent neural networks (RNN) capture temporal data, with the ability to grab contextual information within a sequence of arbitrary length. Convolutional recurrent neural networks (CRNN) combines the best of both worlds to handle multi-dimensional data as sequences.

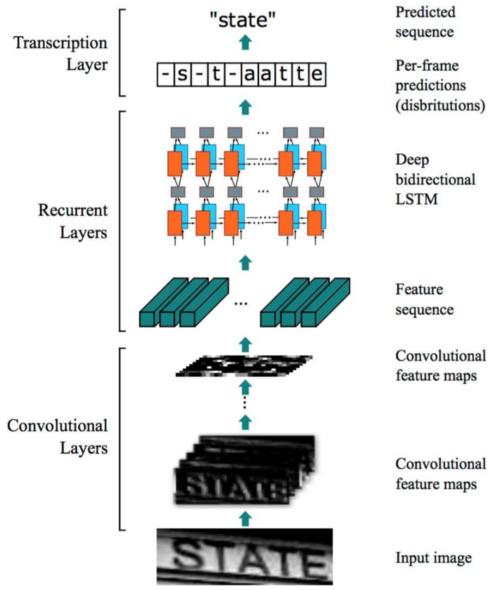


Figure 1: Network architecture. The architecture consists of three parts: 1) convolutional layers, which extract a feature sequence from the input image; 2) recurrent layers, which predict a label distribution for each frame; 3) transcription layer, which translates the per-frame predictions into the final label sequence. [2]

2.2 Training

The CRNN was trained on data coming from various types of Venetian handwritten documents. About 23000 image segments containing 55000 Venetian names of persons and places were manually transcribed by archivists, trained to read such kind of handwritten script, during an annotation phase that lasted 2 years (Fig. 2). Image segments are used in order to reflect only the performance of the transcription system, without introducing possible errors from the segmentation process. Thus, the segmentation step is not part of the proposed experiment. The set was randomly split into training and testing set and the content of the image segments ranges from one to several words (Tab.1).

Set	# images segments	# total words	size of vocabulary
Training set	20712	48628	8848
Testing set	2317	5559	2157
Full set	23029	54187	9429

Table 1: Datasets used

2.3 Results

The performance of the system was evaluated using the Character Error Rate (CER) measure on the test set. Several experiments were performed using different sets of

Figure 2: A few samples of training data image segments

characters (called 'Alphabet' hereafter) and resulted in one model per Alphabet. The numerical results are shown in Tab. 2.

On this dataset, the transcription system has a CER below 10%, which is usually

Alphabet	Set of characters	# image segments	CER
Capital-lowercase-symbols	A-Za-z'.,: -=	24035	0.089
Capitals-lowercase-digits-symbols	A-Za-z0-9'.,:; -_ =()[]/	96198	0.045
Digits	0-9	72326	0.013

Table 2: The Character Error Rate (CER) for each Alphabet

sufficiently good to be able to search for entities in documents.

2.4 Evaluation against human performance

We were interested in comparing the performance of automatic transcription with the average human transcription error. We conducted an experiment on Crowdflower's platform (now Figure Eight), where Italian speaking persons were paid to transcribe image segments of the testing set. The contributors had to decipher a few units before being able to start the survey and during the experiment some of their transcriptions were evaluated. There were 103 evaluation questions that allowed to separate low accuracy contributors' answers from reliable ones. Each image segment was transcribed at least three times, and in total 11'727 units were transcribed. Only the answers of contributors maintaining at least 60% accuracy throughout the experiment and who transcribed at least 50 units were taken into account for the analysis. This resulted in a total of 8'674 valid transcriptions to analyse.

We compare the performance of the system and the amateur transcribers in Tab.3 and Fig.3 (onesample t-test, $p < 0.005$). It is clear from the graphs that the CRNN system has a better CER than the human average on this dataset, and only a few contributors have lower or comparable performance to the system, but is not yet as good as the expert. It is interesting to notice that the performance of the best amateur transcriber almost doubles when capital letters and punctuation are not considered (case 3) whereas the CRNN makes little improvement. Indeed, although the system has inferred some sort of weak language model, we have seen it producing unlikely transcriptions whereas the best contributor uses its knowledge of Italian proper nouns to deduce the correct transcription when some characters are difficult to read.

The developed system shows promising results to make possible the textual search

Case	CER	
	CRNN	contributors
0 : No modifications (Fig.3a)	0.0804	0.1328
1 : Capital letters replaced by lowercase (Fig.3b)	0.0768	0.1137
2 : All punctuation removed (Fig.3c)	0.0766	0.1241
3 : Combination of Case 1 and Case 2 (Fig.3d)	0.0718	0.1047

Table 3: Comparison of Character Error Rates (CER) considering different formatting cases of the transcriptions for the automatic system and the mean of the contributors (ground-truth and predictions are formatted in the same way)

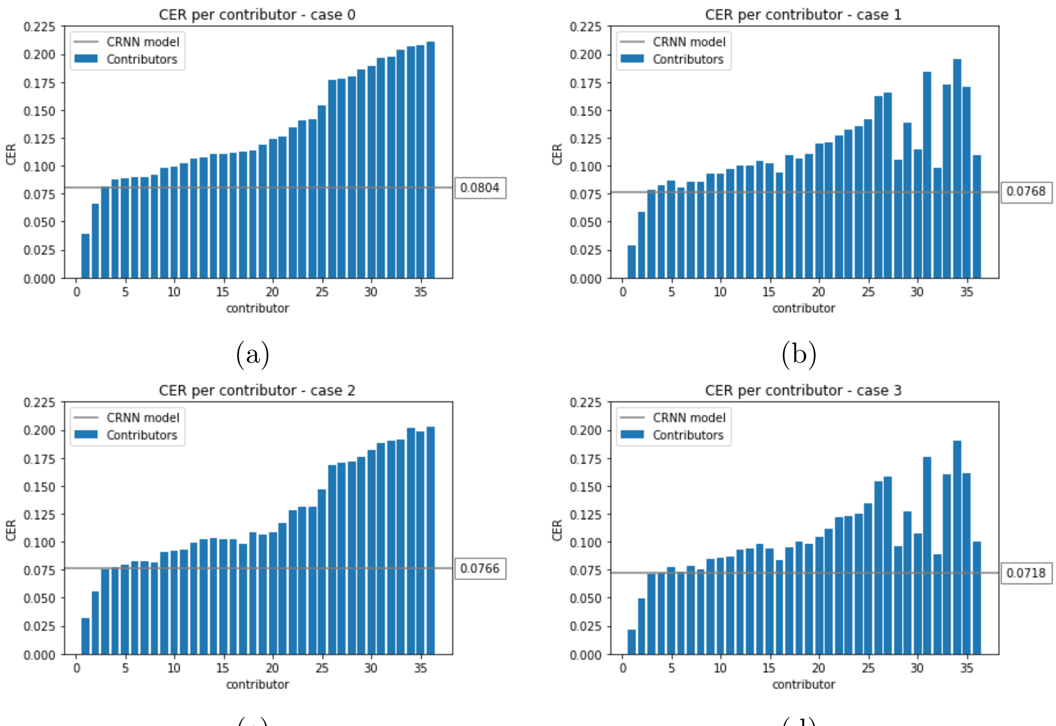


Figure 3: Character Error Rate per contributor for different cases (refer to Tab.3).

on digitized handwritten documents and opens up new prospects for automatic massive indexing. We showed that the system had lower Character than the human average, thus being sufficiently reliable to use for searching purposes. This demonstrates that if the usage of expert transcribers is not possible, for instance for very large collections, the automatic transcriber can achieve results at least as good as amateur transcribers, thus accelerating the process and allowing to focus the expert's resources on the most difficult collections, handwritings or cases.

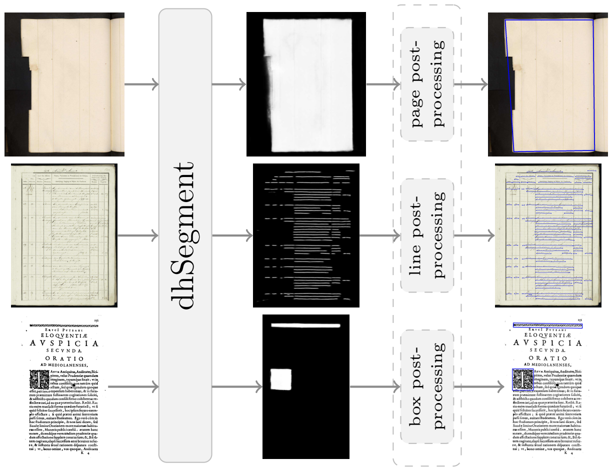


Figure 4: Overview of the system. From an input image, the generic neural network (dhSegment) outputs probabilities maps, which are then post-processed to obtain the desired output for each task.

3 Document segmentation using pixel-wise segmentation

In recent years there have been multiple successful attempts tackling document processing problems separately by designing task specific hand-tuned strategies. We argue that the diversity of historical document processing tasks prohibits to solve them one at a time and shows a need for designing generic approaches in order to handle the variability of historical series. dhSegment [3], a deep-learning approach for document segmentation is our contribution towards this goal.

3.1 Architecture

dhSegment is a general and flexible architecture for pixel-wise segmentation related tasks on historical documents.
The system is based on two successive steps which can be seen in Figure 4:

- The first step is a fully convolutional neural network which takes as input the image of the document to be processed and outputs a map of probabilities of attributes predicted for each pixel. Training labels are used to generate masks and these mask images constitute the input data to train the network.
- The second step transforms the map of predictions to the desired output of the task.

The architecture of the network is depicted in Figure 5. dhSegment is composed of a contracting path (we reuse the terminology 'contracting' and 'exapanding' paths of [4]), which follows the deep residual network ResNet-50 [5] architecture (yellow blocks), and a expansive path that maps the low resolution encoder feature maps to full input resolution feature maps. Each path has five steps corresponding to five feature maps' sizes

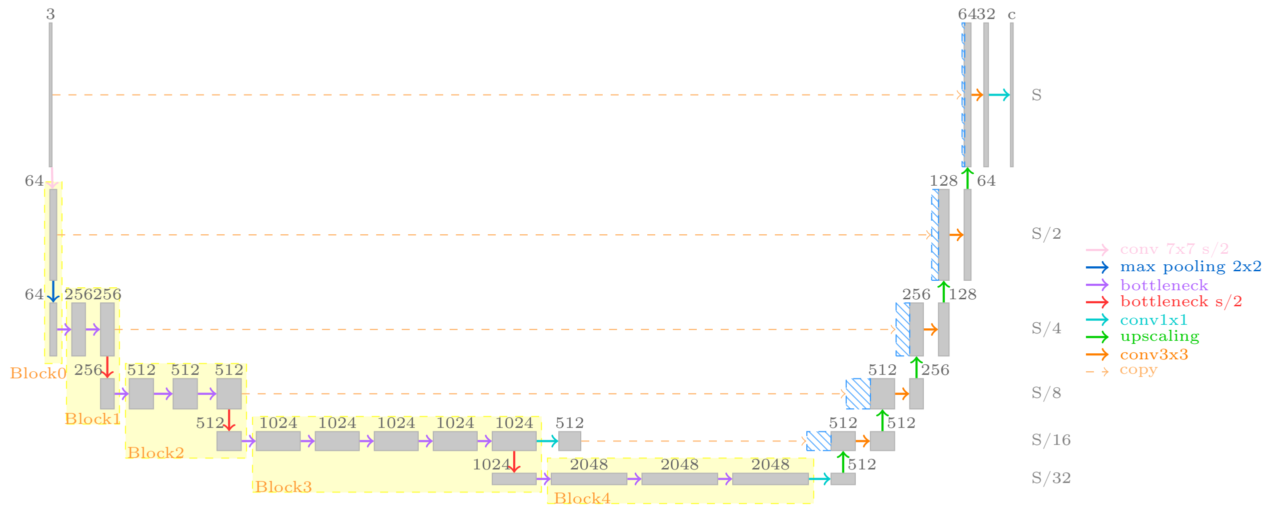


Figure 5: Network architecture of dhSegment. The yellow blocks correspond to ResNet-50 architecture which implementation is slightly different from the original in [5] for memory efficiency reasons. The number of features channels are restricted to 512 in the expansive path in order to limit the number of training parameters, thus the dimensionality reduction in the contracting path (light blue arrows). The dashed rectangles correspond to the copies of features maps from the contracting path that are concatenated with the up-sampled features maps of the expanding path. Each expanding step doubles the feature map's size and halves the number of features channels. The output prediction has the same size as the input image and the number of features channels constitute the desired number of classes.

S, each step i halving the previous step's feature maps size.

The contracting path uses pretrained weights as it adds robustness and helps generalization. It takes advantage of the high level features learned on a general image classification task (ImageNet [6]).
The expanding path is composed of five blocks plus a final convolutional layer which assigns a class to each pixel. Each deconvolutional step is composed of an upscaling of the previous block feature map, a concatenation of the upscaled feature map with a copy of the corresponding contracting feature map and a 3x3 convolutional layer followed by a rectified linear unit (ReLU) [7].
The architecture contains 32.8M parameters in total but since most of them are part of the pre-trained encoder, only 9.36M have to be fully-trained.

3.2 Post-processing

Our general approach to demonstrate the effectiveness and genericity of our network is to limit the post-processing steps to simple and standards operations on the predictions.

Thresholding: Thresholding is used to obtain a binary map from the predictions output by the network. If several classes are to be found, the thresholding is done class-wise.

The threshold is either a fixed constant ($t \in [0, 1]$) or found by Otsu's method [8].

Morphological operations: Morphological operations are non-linear operations that originate from mathematical morphology theory [9]. They are standard and widely used methods in image processing to analyse and process geometrical structures. The two fundamental basic operators, namely the erosion and dilation, can be combined to result in opening and closing operators. We limit our post-processing to these two operators applied on binary images.

Connected components analysis: In our case, connected components analysis is used in order to filter out small connected components that may remain after thresholding or morphological operations.

Shape vectorization: A vectorisation step is needed in order to transform the detected region into a set of coordinates. To do so, the blobs in the binary image are extracted as polygonal shapes. In fact, the polygons are usually bounding boxes represented by four corner points, which may be the minimum rectangle enclosing the object or quadrilaterals. The detected shape can also be a line and in this case, the vectorization consists in a path reduction.

3.3 Training

The training takes advantage of on-the-fly data augmentation strategies, such as rotation, scaling and mirroring, since data augmentation has shown to result in less original data needs. The images are resized so that the total number of pixels lies between $6 \cdot 10^5$ and 10^6. Images are also cropped into patches of size 300×300 in order to fit in memory and allow batch training, and a margin is added to the crops to avoid border effects.
In practice, setting up the training process is rather easy. The training parameters and choices are applicable to most document segmentation tasks and the only parameter that needs to be chosen is the resizing size of the input image. Indeed, the resolution of the input image needs to be carefully set so that the receptive field of the network is sufficiently large according to the type of task.
All trainings and inferences run on a Nvidia Titan X Pascal GPU. Thanks to the pretrained weights used in the contracting path of the network, the training time is significantly reduced. During the experiments, we also noted that the pretrained weights seemed to help regularization since the model appeared to be less sensitive to outliers.

3.4 Experiments

In order to investigate the performance of the proposed method and to demonstrate its generality, dhSegment is applied on different tasks related to document processing. Three tasks consisting in page extraction, baseline detection and document segmentation are evaluated and the results are compared against state-of-the art methods. For each task, the reported results are averaged over 5 runs. See [3] for the complete details of each experiment.

3.4.1 Page extraction

Images of digitized historical documents very often include a surrounding border region, which can alter the outputs of document processing algorithms and lead to undesirable results. It is therefore useful to be able to extract the page document from the digitised image. We use the dataset proposed by [10] to apply our method and compare our results to theirs in Table 4. Our method achieves very similar results to human agreement.

Table 4: Results for the page extraction task (mIoU)

Method	cBAD-Train	cBAD-Val	cBAD-Test
Human Agreement	-	0.978	0.983
Full Image	0.823	0.831	0.839
Mean Quad	0.833	0.891	0.894
GrabCut [11]	0.900	0.906	0.916
PageNet [10]	0.971	0.968	0.974
dhSegment (quads)	$0.98 \pm 6 \cdot 10^{-4}$	$0.98 \pm 8 \cdot 10^{-4}$	$0.98 \pm 8 \cdot 10^{-4}$

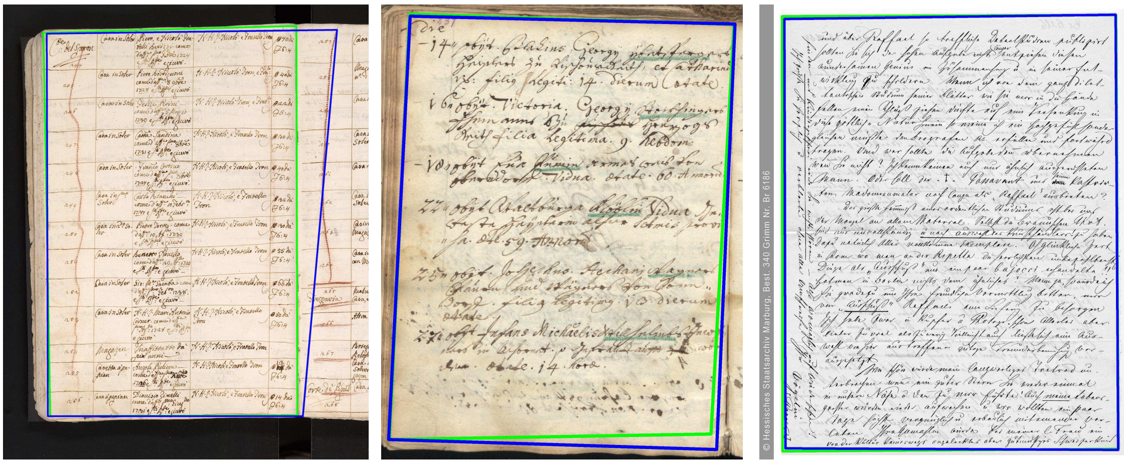


Figure 6: Example of page detection on the cBAD-test set. Green rectangles indicate the ground-truth pages and blue rectangles correspond to the detections generated by dhSegment. The first extraction is inaccurate because part of the side page is also detected. The second one is slightly inaccurate according to the groundtruth but the entire page is still detected. The last example shows a correct detection.

3.4.2 Baseline detection

Text line detection is a key step for text recognition applications and thus of great utility in historical document processing. A baseline is defined as a "virtual line where most characters rest upon and descenders extend below". The READ-BAD dataset [12] has been used for the cBAD: ICDAR2017 Competition [13] and the results can be seen in Tab. 5 and Fig. 7.

Table 5: Results for the cBAD : ICDAR2017 Competition on baseline detection [13] (test set)

Method	Simple Track			Complex Track		
	P-val	R-val	F-val	P-val	R-val	F-val
LITIS	0.780	0.836	0.807	-	-	-
IRISA	0.883	0.877	0.880	0.692	0.772	0.730
UPVLC	0.937	0.855	0.894	0.833	0.606	0.702
BYU	0.878	0.907	0.892	0.773	0.820	0.796
DMRZ	0.973	0.970	0.971	0.854	0.863	0.859
dhSegment	0.88±.023	0.97±.003	0.92±.011	0.79±.021	0.95±.005	0.86±.011
ARU-Net [14]	0.977	0.980	0.978	0.926	0.918	0.922

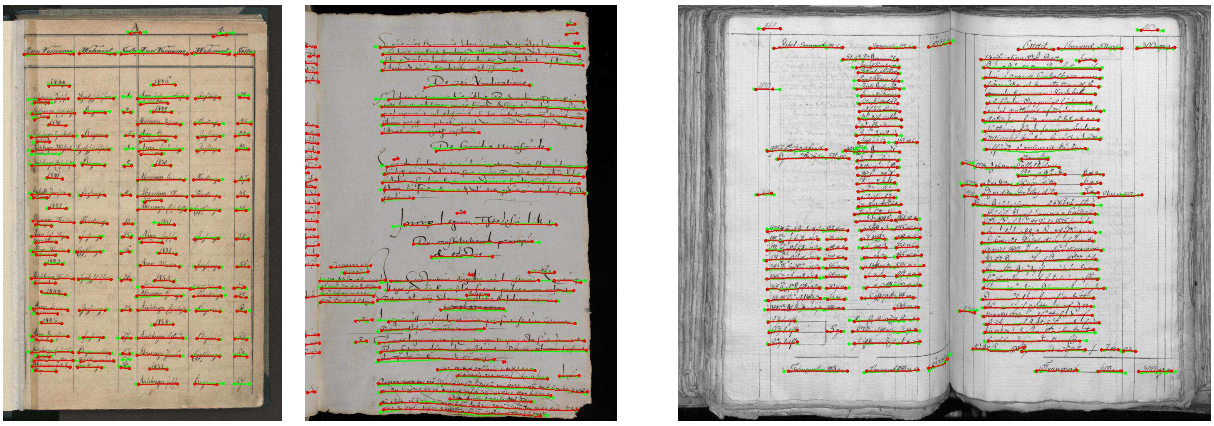


Figure 7: Examples of baseline extraction on the complex track of the cBAD dataset. The ground-truth and predicted baselines are displayed in green and red respectively. Some limitations of the simple approach we propose can be seen here, for instance detecting text on the neighbouring page (right), or merging close text lines together (left and middle). These issues could be addressed with a more complex pipeline incorporating, for instance, page segmentation (as seen in Section 3.4.1) or by having the network predict additional features, but this goes beyond the scope of this paper.

3.4.3 Document layout analysis

Document Layout Analysis refers to the task of segmenting a given document into semantically meaningful regions. In the experiment, we use the DIVA-HisDB dataset [15] and perform the task formulated in [16]. In this task, the layout analysis focuses on assigning each pixel a label among the following classes : text regions, decorations, comments and background, with the possibility of multi-class labels (e.g a pixel can be part of the main-text-body but at the same time be part of a decoration). Our results are compared with the participants of the competition in Table 6. While using the same

Table 6: Results for the ICDAR2017 Competition on Layout Analysis for Challenging Medieval Manuscripts [16] - Task-1 (IoU)

Method	CB55	CSG18	CSG863	Overall
System-1 (KFUPM)	.7150	.6469	.5988	.6535
System-6 (IAIS)	.7178	.7496	.7546	.7407
System-4.2 (MindGarage-2)	.9366	.8837	.8670	.8958
System-2 (BYU)	.9639	.8772	.8642	.9018
System-3 (Demokritos)	.9675	.9069	.8936	.9227
dhSegment	.974±.001	.928±.002	.905±.007	.936±.004
dhSegment + Page	.978±.001	.929±.002	.909±.006	.938±.004
System-4.1 (MindGarage-1)	.9864	.9357	.8963	.9395
System-5 (NLPR)	.9835	.9365	.9271	.9490

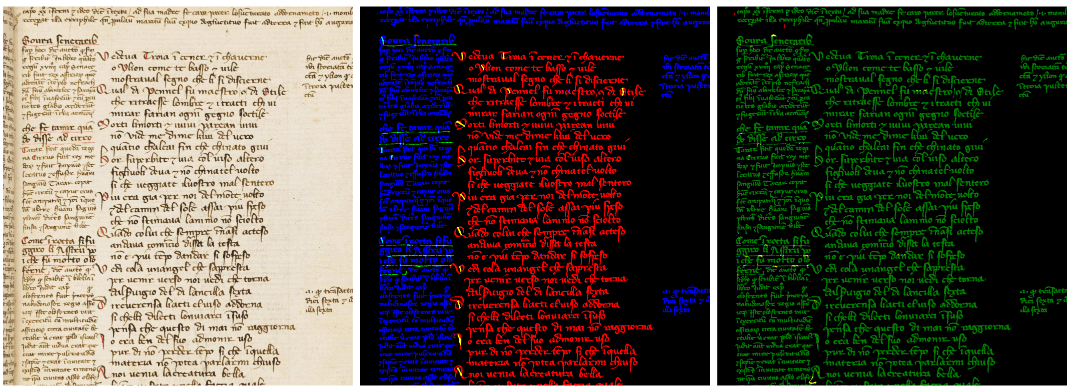


Figure 8: Example of layout analysis on the DIVA-HisDB test test. On the left the original manuscript image, in the middle the classes pixel-wise labelled by the dhSegment and on the right the comparison with the ground-truth (refer to the evaluation tool[1]for the signification of colors)

network and almost the same training configurations, the results we obtained on the three different tasks evaluated are competitive with the state-of-the-art. Besides the genericity and flexibility of the approach, we can also highlight the speed of the training (less than an hour in some cases), as well as the little amount of training data needed, both thanks to the pre-trained part of the network.

[1]https://github.com/DIVA-DIA/DIVA_Layout_Analysis_Evaluator

4 Cadastral computing

Among all the diverse typologies of administrative systems, the fiscal-cadastral sources retracing the ownership of lands are undoubtedly the richest and, in a sense, the most coherent records. The cadastres established during the first years of the 19th century cover a large part of Europe. For many cities they give one of the first geometrical surveys, linking precise parcels with identification numbers. These identification numbers point to register lines with the names of the parcel's owners (Fig. 9). As the Napoleonic cadastres include millions of parcels, it therefore offers a detailed snapshot of large part of Europe's population at the beginning of the 19th century [17].
Two challenges need to be solved in order to be able to automatically process cadastral sources: (1) developing algorithms capable of robustly segmenting maps into parcels and administrative tables into cells (2) developing solutions for transcribing handwritten text containing people or places mentions and identification numbers.
Previous year's work focused on the cadastre map processing, namely extracting parcel's geometries and localizing and transcribing identification numbers.
The method has since then been improved, based on our progress on document segmentation (see Section 3). Also the evaluation was extended since the full cadastre has been annotated manually. Regarding the register documents, dhSegment enabled the extraction of text lines and columns detection. Moreover the transcription system presented in Section 2 allowed to transcribe the extracted text lines.

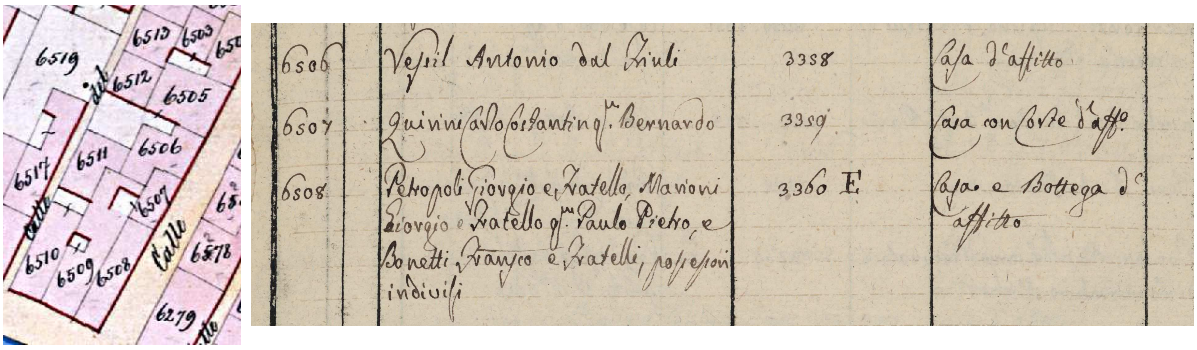


Figure 9: Identification numbers in maps and in the register documents

4.1 Cadastre maps

4.1.1 Method

The automatic processing of the cadastre maps aims at extracting the parcels as geometrical shapes and also at transcribing the parcel's identification numbers. It is composed of three main steps:

1. Training of the deep neural network on manually annotated data
2. Segmentation of the maps into meaningful regions and objects
3. Transcription of the identification numbers.

The segmentation network is the fully convolutional neural network presented in Section 3 and in [3]. The network is trained to extract the parcel contours and text using annotated data from the Venetian cadastre (Fig. 10).

The transcription network is the convolutional recurrent neural network (CRNN) de-

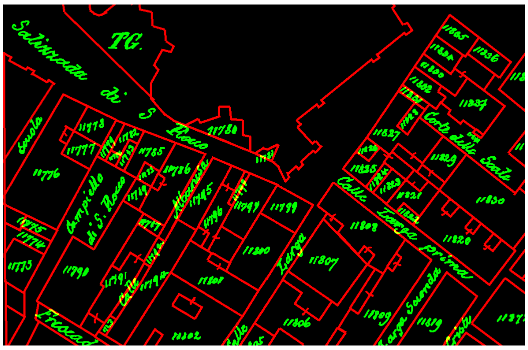
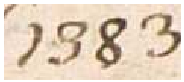


Figure 10: Sample of training data for cadastre maps segmentation. Parcels contour are in red, text is in green.

scribed in Section 2. The network is trained on samples of numbers from the Venetian archives and on numbers synthetically generated with MNIST digits [?] (Fig. 11).

(a) synthetically generated numbers



(b) numbers form the Venetian archives

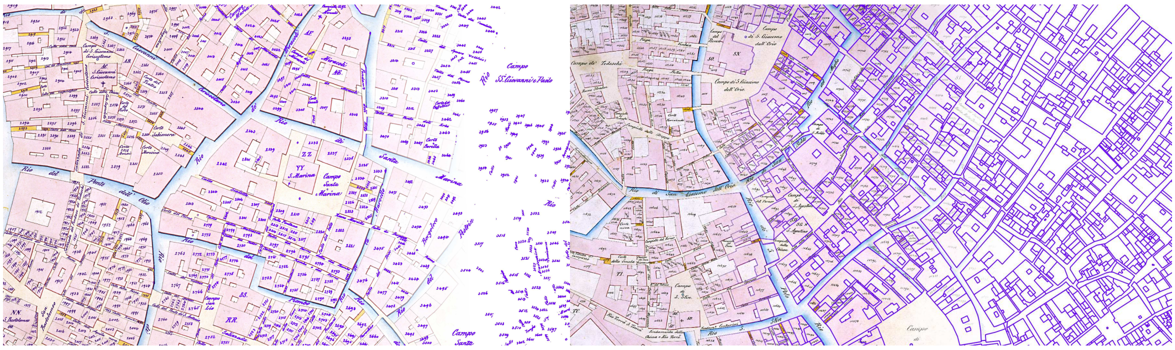
Figure 11: Example of training data for the transcription system.

The segmentation model obtained after the training is able to predict the parcel contours and text region at pixel level (Fig. 12). Then, similar operations to the ones of the previous workflow are used. Watershed by flooding algorithm [?] is applied on parcel contours predictions, which allows the extraction of parcel objects as polygonal shapes. Text regions are cropped, horizontally aligned and converted into grayscale image segments for further processing by the transcription system. The image segments containing text are fed to the transcription network, which outputs a prediction of a number. Each transcription is then linked to its corresponding parcel.

The contours of the parcels (whether they contain an identifier number or not) are saved as polygonal shapes and are exported into JSON format. In our case, since the images have previously been georeferenced, the coordinates are exported as geographical coordinates and can therefore directly be imported in any geographic information system.

4.1.2 Results

Two evaluations are performed in order to assess the performances of the system: the geographical accuracy of the extracted parcels and the transcription of the identification numbers. The evaluation process is similar to last years' but this time we are able to evaluate it on the entire cadastre since it has been entirely annotated.



(a) text extraction (b) contour extraction

Figure 12: Output of the segmentation network (overlay in purple)

Geometrical evaluation

The number of geometrical shapes extracted and manually annotated are listed in Tab. 7.

The quality of the parcel's extraction is evaluated by measuring the intersection over union (IoU) between the geometries produced automatically and almost 17000 manually annotated shapes.

Geometries	Number
Geometries extracted automatically	31 342
Geometries remaining after filtering	28 711
with ID number	18 138
Manually annotated geometries	16 946
with ID number	15 634

Table 7: Total number of geometries in the automatic extraction and manual annotation. The first three rows relate to automatically extracted parcels, the two last rows show the statistics for manually annotated parcels.

Precision and recall with three different IoU thresholds 0.5 (acceptable), 0.7 (good), 0.9 (excellent) are reported in Tab. 8. The recall value shows that a large majority of parcels are extracted. The low precision value is mainly due to the incorrect extraction of streets, squares, canals, etc. that are currently not filtered out (example in Fig. 14).

IoU	Correct parcels	Precision	Recall
t=0.5	15999	0.557	0.944
t=0.7	15292	0.533	0.902
t=0.9	14440	0.503	0.852

Table 8: Evaluation of the geometrical shape extraction with different Intersection over Union (IoU) thresholds

Figure 13: Visualization of the results of the automatic extraction of parcels. The red rectangle indicates the parcels used as training data for the segmentation network.

Transcription evaluation

We assess the performance of the transcription of parcel's identifier numbers by computing the number of correct predictions and report the precision and recall values in Tab. 9. The current method assumes that the identifiers are located within the parcel, thus, identifiers partially or completely outside the geometrical shape are not correctly transcribed (Fig. 15), resulting in a lower recall.

In order to increase the precision value and since we can assume that spatially close parcels will have numerically close identifiers, we tried to discard false predictions by determining if a transcription was 'plausible' or not, using information from its spatial neighbourhood. Thus, a transcription is considered as an outlier if the (numerical) difference between the predicted number and the median of its 5 neighbouring transcriptions is greater than 10. This results in a significant increase in precision (up to 93%), but at the expense of a decrease in recall.

	Correct transcriptions	Precision	Recall
Transcriptions	11101	0.612	0.710
Transcriptions after outlier detection	8070	0.927	0.516

Table 9: Evaluation of the transcriptions of parcel's identifiers numbers

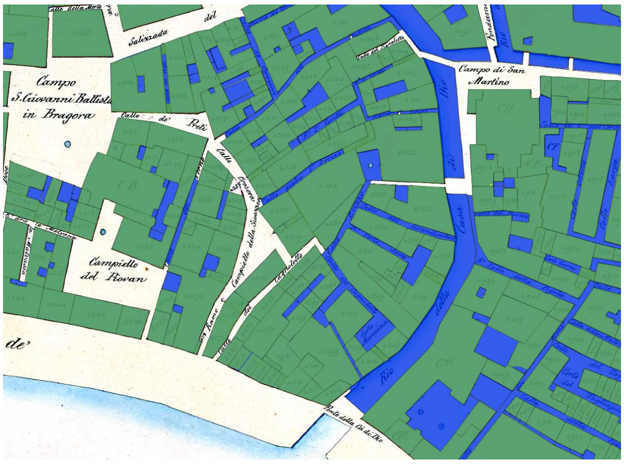


Figure 14: Example of false extraction of streets and canals (in blue)

Figure 15: Example of identifiers numbers outside or partially outside the parcel

4.2 Register documents

The register documents (Fig. 9, on the right) are table documents containing identifiers numbers, the name of the owners and other information relative to the land property. There are approximately 1500 pages Three steps constitute the processing of the register: (1) segmentation of the table layout (page, column, headers, ...); (2) detection of the text lines; (3) transcription of the text.
The layout segmentation uses dhSegment (Sec. 3, [3]). Page extraction model presented in section 3.4.1 and the line extraction model presented in section 3.4.2. A model is trained for column and header extraction for this particular collection.
Once the lines are extracted, they are fed to the transcription system presented in section 2. The model used for the transcription of the registers is trained on data from other Venetian documents with similar handwriting (same data used in section 2.2 with additional text segments containing most common abbreviations).
It is then possible to cluster the text lines by column and therefore by type of information. It is particularly interesting in our case to group and label the text lines corresponding to the identifier numbers, in order to be able to link them to the identifiers in the cadastre maps. By identifying the text lines belonging to the identifiers group, we also are able to apply more efficient transcription model, specialized in digits recognition, in order to improve the predictions.

The pipeline for linking parcel's identifiers to their corresponding entry in the registers has been tested end-to-end, since we were able to automatically link the information of both documents, but is still on going work. Future work will focus on improving and automatically correcting the transcriptions of the identifiers and evaluating the extent of the information linking.

5 Time Machine : Diamond platform

In the context of the Time Machine European FET Flagship and following the progress made during this last year, a prototype search engine, Diamond[2], was released. Diamond includes search functionalities based on machine transcription of handwriting, iconographic search based on deep learning technology and a spatiotemporal interface for exploring the structure of cities in the past. In particular, part of the cadastral computing work (section 4) is included in the platform. This prototype is intended to bootstrap a collective effort towards the building of an open-source, free and scalable search engine. It permits now to search through a couple of millions documents but this number is expected to hugely grow in the coming months, as part of the Time Machine project.

[2]https://diamond.timemachine.eu/

References

[1] S. A. Oliveira and F. Kaplan, "Comparing human and machine performances in transcribing 18th century handwritten venetian script," in *Digital Humanities 2018, DH 2018, Book of Abstracts, El Colegio de México, UNAM, and RedHD, Mexico City, Mexico, June 26-29, 2018*, pp. 150–155, 2018.

[2] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[3] S. Ares Oliveira, B. Seguin, and F. Kaplan, "dhsegment: A generic deep-learning approach for document segmentation," in *Frontiers in Handwriting Recognition (ICFHR), 2018 16th International Conference on*, pp. 7–12, IEEE, 2018.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[9] J. Serra, *Image analysis and mathematical morphology.* Academic Press, Inc., 1983.

[10] C. Tensmeyer, B. Davis, C. Wigington, I. Lee, and B. Barrett, "Pagenet: Page boundary extraction in historical handwritten documents," in *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, pp. 59–64, ACM, 2017.

[11] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.

[12] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 351–356, IEEE, 2018.

[13] M. Diem, F. Kleber, S. Fiel, T. Gruning, and B. Gatos, "cbad: Icdar2017 competition on baseline detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 1355–1360, Nov. 2017.

[14] T. Grüning, G. Leifert, T. Strauß, and R. Labahn, "A two-stage method for text line detection in historical documents," *arXiv preprint arXiv:1802.03345*, 2018.

[15] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pp. 471–476, IEEE, 2016.

[16] F. Simistira, M. Bouillon, M. Seuret, M. Würsch, M. Alberti, R. Ingold, and M. Liwicki, "Icdar2017 competition on layout analysis for challenging medieval manuscripts," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 1361–1370, IEEE, 2017.

[17] P. c. Clergeot, *Cent millions de parcelles en France : 1807, un cadastre pour l'Empire*. Editions Publi-Topex, 2007.