

French vital records data gathering and analysis through image processing and machine learning algorithms

Cyprien Plateau–Holleville^{*1}, Enzo Bonnot^{*1}, Franck Gechter^{1,2}, Laurent Heyberger^{1,3}

* Both authors have contributed the same to this work

¹ Univ. Bourgogne Franche-Comte, UTBM, F-90010, Belfort, France

² CIAD (UMR 7533) and LORIA-MOSEL (UMR 7503), Université de Lorraine

³ FEMTO-ST Recits (UMR 6174)

Abstract

Vital records are rich of meaningful historical data concerning city as well as countryside inhabitants that can be used, among others, to study former populations and then reveal the social, economic and demographic characteristics of those populations. However, these studies encounter a main difficulty for collecting the data needed since most of these records are scanned documents that need a manual transcription step in order to gather all the data and start exploiting it from a historical point of view. This step consequently slows down the historical research and is an obstacle to a better knowledge of the population habits depending on their social conditions. Therefore in this paper, we present a modular and self-sufficient analysis pipeline using state-of-the-art algorithms mostly regardless of the document layout that aims to automate this data extraction process.

Keywords

Historical Data - Optical Character Recognition - Handwritten Text Recognition - Machine Learning

INTRODUCTION

Historical serial data gathering is a critical step in many fields that requires to manually deal with text documents damaged by time. The need for contributors transcription teams of sufficient size can thus significantly slow down linked research work and make its cost dramatically rise. Developing automation through Optical Character Recognition (OCR), Handwritten Text Recognition (HTR), and Natural Language Processing (NLP) technologies could then be a real time saver and a strong support for this application.

The OCR and HTR state of art has made great progress since the coming of machine learning as shown by Breuel et al. [2013], Ingle et al. [2019], or Yousef and Bishop [2020] which made available cutting edge and production-ready algorithms to the public. The help provided by the use of those technologies is significant in many fields to automate time-consuming tasks. Some fields such as historical document analysis are, however, still challenging even with the growth of these technologies. This project is a work that aims to focus on providing a set of tools that could be used to facilitate the research effort which can be slowed by the lack of digitised data.

In addition to its potential methodological and practical repercussions on a national scale, the present OCR project for the French vital records is part of a collaborative project between engi-

neering sciences and human and social sciences entitled "Techn'Hom Time Machine" (THTM). Led by researchers from the University of Technology of Belfort-Montbéliard, it focuses on Belfort, a French city located in the north-east of the Franche-Comte region, a few kilometres west of the Swiss city of Basel. The THTM project aims to reconstruct the architectural and technical history, via industrial archaeology, on the one hand, and the demographic and social history of the city's main working-class district, on the other. Belfort indeed experienced very rapid industrialisation after the Franco-Prussian conflict of 1870, which makes it very interesting to study its population from a demographic point of view through the various sources of historical demography: as a boomtown, it does not seem to have experienced a deterioration in living standards during its phase of rapid urbanisation: see Heyberger [2013].

Data provided in this project is composed of unlabelled-scanned images of French vital records, an example of which is displayed in figure 1. These data are hosted in the Belfort archives website available at the following [https URL](#). The French vital records include data that are of interest to historical research, here mainly: the date and place of birth of individuals (neighborhood and street, but also home or hospital), the identity and occupation of the two witnesses to the declaration of birth, the date and place of death of individuals, the occupation of the father and mother of individuals, their sex, and the date and place of marriage of individuals. The use of these data, which had already been started manually, has made it possible to produce some very interesting initial results regarding the longevity of the Belfortains born around 1900 as explained in Haton [2020]. However, here as elsewhere, the development of more automated processing of sources would make it possible to break one of the most important barriers facing cliometry and demographic history.

The gathering of this data can be assisted by tools such as the European project Transkribus as shown by Massot et al. [2019] and Schlagdenhauffen [2020]. The latter gives access to efficient HTR and OCR algorithms through an ergonomic user interface in order to perform the transcription process. This solution is, however, restricted in its charge-free version and use a closed-source HTR engine that can make it much less accessible. Other well-known software such as Tesseract offers efficient OCR algorithms but as explained by Rakshit et al. [2010] and Rakshit and Basu [2010], the engine might be less accurate on handwritten data. Finally, to effectively help humanities research, the need of proper text data extraction is unavoidable. In the current study, the gathering of precise sample information regarding the individuals in question is needed. This knowledge should then be transformed in a queryable shape to enable the use of classic tools by final users. As shown by Dudhabaware and Madankar [2014], this operation can be achieved through NLP algorithms that can perform deep and accurate analysis in the aim of data labelling congregation through syntactical and lexical analysis. The latter will not be addressed since it is beyond the scope of this article.

The lack of a complete pipeline makes the data acquisition task harder for the user and not suitable for historical document analysis. This is why, in this article, we present the bases of an adaptive, modular and extendable analysis pipeline for historical documents based on state-of-the-art deep learning and image processing algorithms that aim to limit most manual interventions. We release the code at the following [https url](#).

This article is structured as follows. The first section is focused on historical data analysis state of the art based on image processing and deep learning algorithms. The proposition details are then explained in the second section, supported by experimental results. Finally, the third section concludes the paper by providing some potential future works.

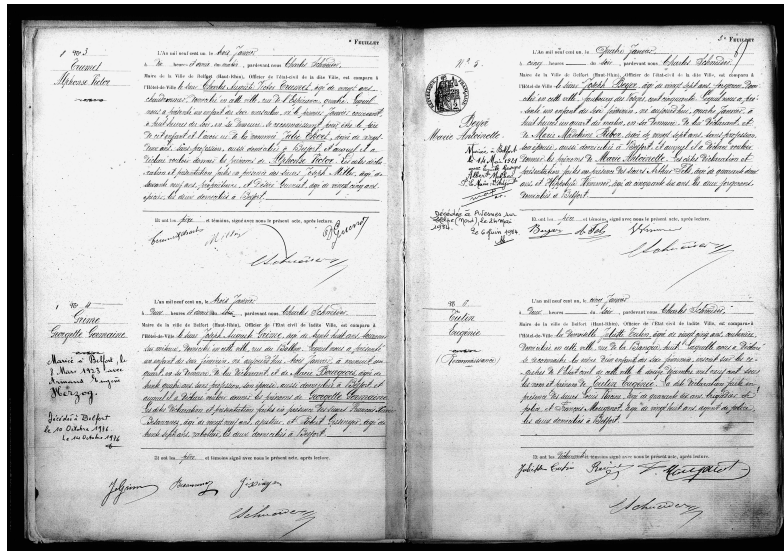


Figure 1: Scanned vital record sample (registres d'état-civil des naissances, 1901, archives of the "Territoire de Belfort" department)

I STATE OF THE ART

1.1 Image preprocessing

The extraction of text information within physical documents can be highly sensitive to the support general aspect. This can be lowered by the use of preprocessing algorithms that aim to erase mostly non-text information inside images to only keep interesting features. This bivariate classification step can be done through automatic thresholding based algorithms as demonstrated by Otsu [1979]. However, this solution can fail to produce proper text segmentation on historical data. Indeed, this kind of data can present a lot more degradation, artefacts, or bleeding than digital or modern data. It is then mandatory to adapt the binarisation algorithm to the specification. Wolf et al. [2002], Gatos et al. [2004] and Bataineh et al. [2011] have shown that adaptive strategies handle more effectively variations within the document by performing the binarization on small parts of the image and adapting the thresholding settings to the current context. This helps to keep text information even if there are high disparities within the document.

1.2 Image segmentation

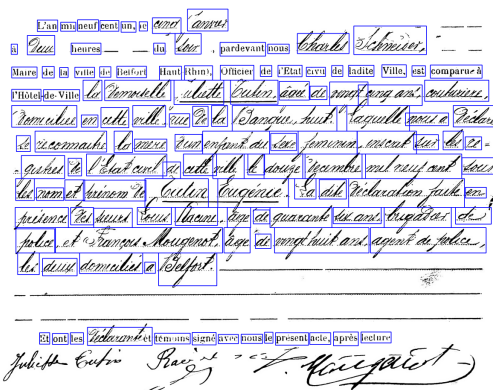


Figure 2: Word level ideal text segmentation of a single paragraph (Blue: Word-level bounding boxes)

Handwritten text segmentation is an unavoidable step of every handwritten text recognition process and its goal is to provide precise information of the text position inside the analyzed document as shown in figure 2. Bad results from the segmentation will have a negative impact on the recognition. Therefore many different methods have already been implemented and documented. These approaches will divide the text at various scales, the most common being line and word level.

Many of the latest line segmentation techniques use a machine-learning based approach and especially convolutional neural networks allowing to get better results with a much more stable process. Renton et al. [2017] presented their solution using these technologies which give significantly more precise boxes for the extracted lines avoiding them to overlap over. However this method is tested on rather simple data and results on more complex data, as the one treated here, are quite unsure. Concerning word segmentation which is what we are aiming at for the segmentation task, Huang and Srihari [2008] proposed a method processing text lines to extract the words. This neural network reaches very good efficiency assuming that the lines given in input are perfectly segmented. Combining the two previous methods can be very error-prone since it is a cascading process. For this reason, a robust method that would directly extract the words is preferable. Zhou et al. [2017] developed a system for scene text recognition. This tool is trained to detect text in challenging environments like a natural scene or complex background. Hence adapting it to the context of these vital records would be a suitable option.

1.3 OCR

OCR has been a trending topic over the past few years for several reasons. It allows automatic processing of many documents and can therefore accelerate many tasks that are time-consuming and more error-prone when realised by a human. Smith [2007] developed Tesseract, which is a complete OCR pipeline, showing what can be achieved in terms of text recognition. One of the main subfields of OCR is HTR which is very complex since almost no handwriting style is the same and also the lack of structure that can be encountered over many handwritten documents. Therefore HTR is a very important part of the process to solve our problem and several methods have been developed to do so.

Granet et al. [2018] presented an approach for handwriting recognition on historical documents without having to manually establish the ground truth of the document collection studied. The model used is instead trained on data that shares common characteristics with the one to recognise. This solution has been tested on 18th-century handwritten documents in French and Italian and gives interesting results that are, however, insufficient for our scenario. Romero et al. [2011] has proven that the use of hidden Markov models can be a solution to historical handwritten text recognition. The solution described by de Sousa Neto et al. [2020] is a new architecture for HTR that is giving very good results over many datasets including historical handwritten records. It has been tested with English and French handwritten documents and performs the best when the input data is at word level with a character error rate of around 2.5% and a word error rate of around 8.5%. This method then seems well adapted to our context since the data to be recognised, which will be provided to the network, are segmented by words.

1.4 Ground truth creation

To train deep learning models, one needs to create a consequently large dataset that fits with production data characteristics. This task can be heavily time-consuming as a result of the required manual operations. For this reason, solutions to accelerate this process has been developed such as the one presented by Fischer et al. [2010]. This method presents a tool for semi-automatic ground truth creation. It covers this process from segmentation to annotation with alignment

to its corresponding segmented area. Nevertheless, this approach would not fit perfectly our context since it requires the use of external software during the process but also because the segmentation method cannot be modified to include the one chosen for our solution.

II PROPOSITION DETAILS

2.1 General overview

The main issue of this subject is to create realistic and production-based datasets to train deep learning algorithms. In our proposition, we create partially efficient tools that aim to avoid manual work and human intervention. These pipeline sub-processes concern mostly segmentation and OCR node as we can see in figure 3. This article will only present our work on the text segmentation and the text recognition parts of the pipeline.

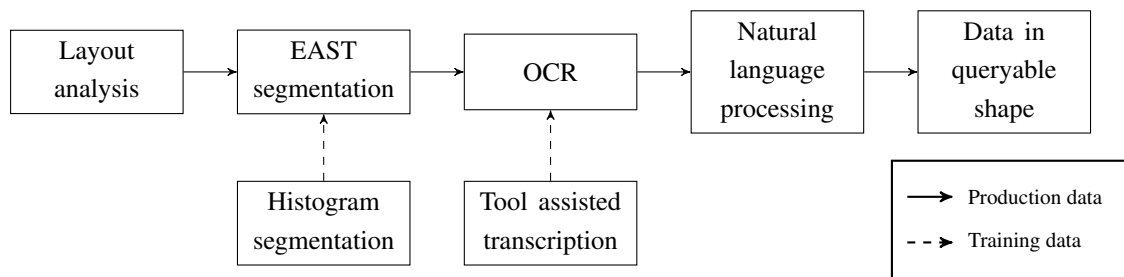


Figure 3: Data acquisition pipeline

2.2 Data presentation

Most of the time, samples are divided into four parts themselves giving information about one person. These are built based on partially structured positional rules which make layout-based analysis strategy hard to set up. The main issue in these archives survey is the presence of irregular types of writing in a single sample. Indeed, the original document was prepared with typewritten standard text that was then handily filled by potentially more than one person. These script changes make line shapes uneven and arduous to properly evaluate through basic algorithms.

2.3 Text Segmentation

Our text segmentation strategy is divided into two parts, one for the training data creation process, the second for the production pipeline process. The need for precision and robustness to deal with partially unstructured layout makes the use of machine learning algorithms unavoidable. A dataset, therefore, has to be produced from accurate production-based data. This step can be done through manual work, however, this task can be highly time-consuming when only a small workforce is available. The creation of dedicated tools aiming to help this task is a good trade-off that enables strong time savings and requires a lot fewer contributors to establish training sets. This might also be applied for the segmentation since text words bounding box annotation can be redundant and error-prone for the contributor. This part will then present our tools that aim to make the annotation steps easier and how it interacts with the learning phase.

2.3.1 Image preprocessing

The input data of the process is raw scanned grayscale images containing scanner artefacts and frames. This initial condition makes the segmentation process harder due to background noise. Therefore, several preprocessing steps are applied whose goal is to enhance text while reducing at most non-text parts. To achieve this, the use of the background detection based on the process

described in figure 4 allows to robustly finding the document outline on our dataset. Then, a gamma correction is performed to improve the general luminosity of the image.

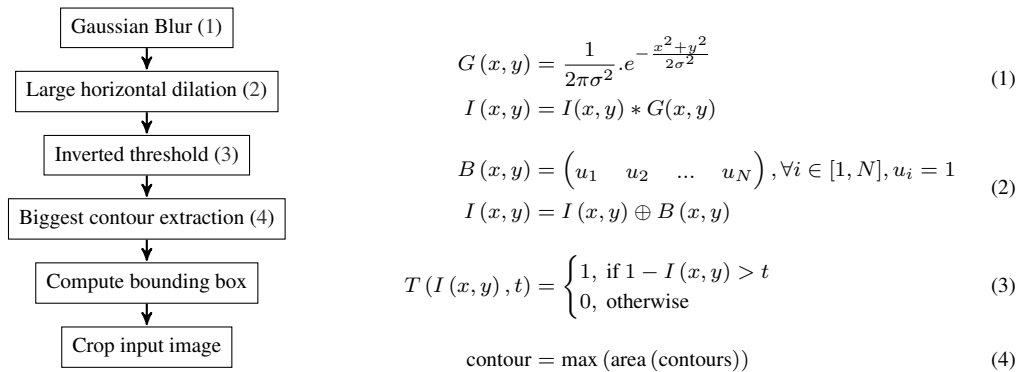


Figure 4: Scanner background extraction process

The classification of both text and background is then made based on a binarisation step. The latter is a challenge in this application since it needs to mostly keep text without any degradation while removing any other data. The nature of the raw image makes the use of basic and global thresholding algorithm unusable due to background irregularities. We, therefore, apply an adaptive local binarization that combines many benefits for this use case¹. As displayed in figure 5, the main property of this method is to apply a threshold on image parts 5b, that are then used to build a mask 5c, which finally allows to segment interesting information of the image 5d. Its properties are particularly well suited to text segmentation since the contrast of the image can be variable. The latter would cause parts of the text to be wiped out by a standard thresholding strategy. This method mainly allows the erasure of paper-related data and noise, while mostly retaining textual information as shown in the binarization process in figure 5. The quality of the segmentation is, therefore, less linked to the complete image features but rather to local information which provides accurate and robust results throughout its use on our dataset.

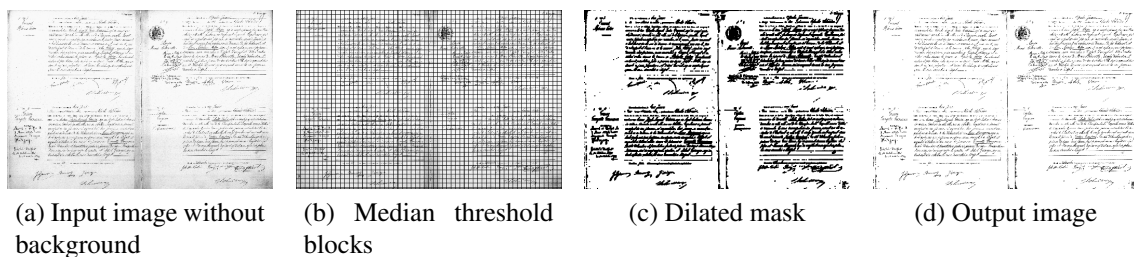


Figure 5: Adaptive local binarization process

This method requires, however, to parameterise the size of the blocks that leads it to be less robust to strong image changes since it will be linked to input image properties.

Finally, figure 6 shows the result of the full preprocessing pipeline and illustrate how the chosen adaptive binarization performs compared to the method introduced by Otsu [1979]. The latter can erase more noise than the one we selected, however, letter shapes are less eroded which enables a better readability.

¹We chose the following implementation: [Robust Locally-Adaptive Soft Binarization! That's what I call it.](#)

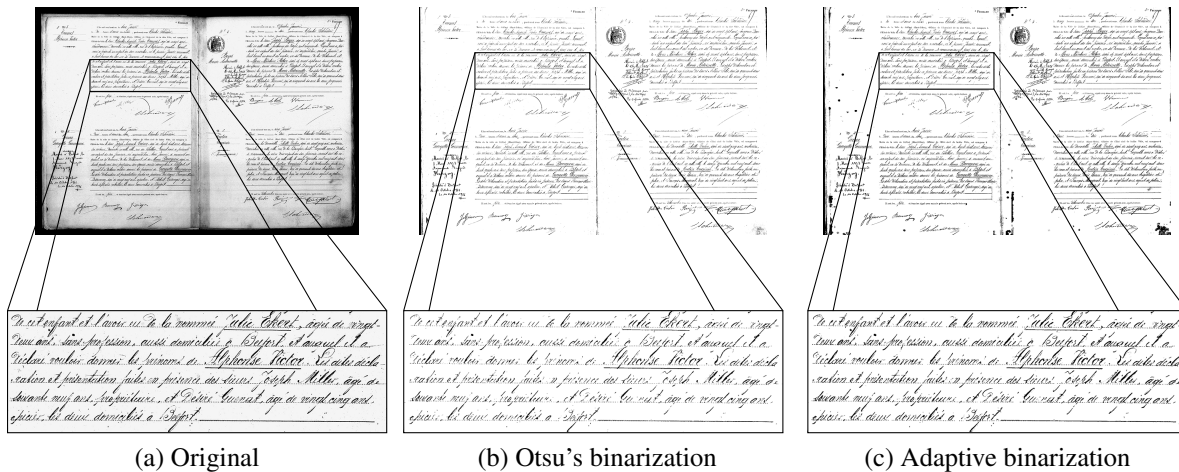


Figure 6: Comparison of Otsu's and adaptive binarization on the full preprocessing pipeline

2.3.2 Histogram projection based segmentation

Once the preprocessing steps are applied to the image, the background removing allows to decrease a lot the analysis complexity since the text is the main remaining information. This enables the development of segmentation heuristics which is the basis of the dataset creation tools that aims to ease the manual work needed for machine learning algorithms. Indeed, in order to provide accurate enough tools and save users' time, the segmentation needs to only require small tweaks during manual inspections. Heuristics developed for this part are then strongly linked to the data since it will only be used in a preproduction process that will not involve the robustness of the final method. This can, however, be easily adapted for other use cases without major pipeline transformation.

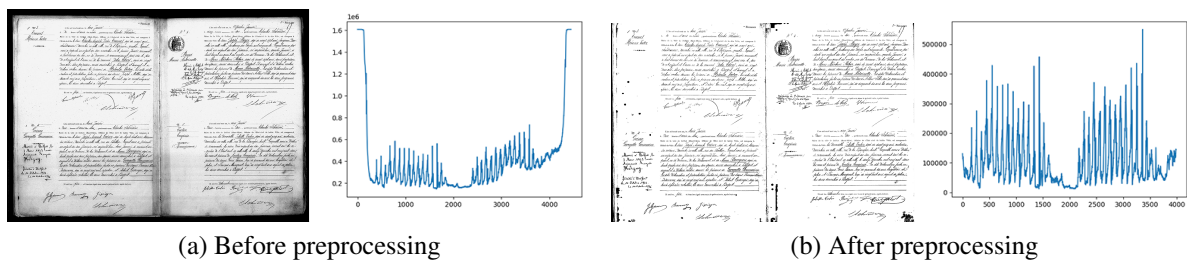


Figure 7: Histogram evolution before and after the preprocessing

As shown in 7, the horizontal histogram of the image is a lot more focused on text information than before. Histogram projections can then be used for text segmentation as demonstrated by Alkalai and Sorge [2013]. The latter demonstrates how, in the case of background-less images, text lines can be analysed accurately on the basis of vertical and horizontal histogram properties. This can be moreover quickly implemented thanks to the simplicity of the process. This is especially needed to easily and accurately build a dataset. Zohrevand et al. [2019] validated this method as well with the support of deskew algorithms which are not needed in the current dataset since lines generally have horizontal levels. The text area's main shapes are then found based on white and black pixel distribution. This method remains, however, highly sensitive to changes in text areas breakdown. Consequently, we chose to guide the general process by computing image histogram projections only on specific image areas derived from the most

common layout as we can see in figure 8. The selection of the less significant element provides finally a way to find paragraph borders.

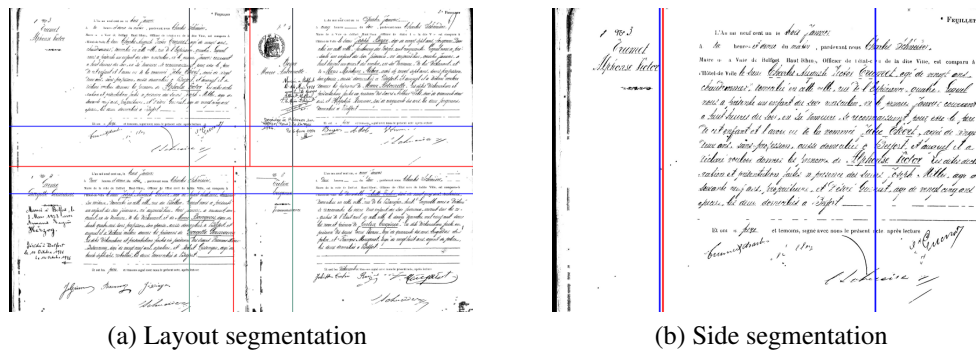


Figure 8: Visualisation of the histogram projection based segmentation (Red : Selected local minimas. Blue, green and black : Histogram projection bounds)

As shown in figure 8b, the side panels of each sample are segmented through the same strategy. This simple method is able, most of the time, to correctly segment paragraphs, however, in the case of less ordered samples, it can be erroneous due to the superposition of the main paragraph and side paragraphs as displayed in figure 9.

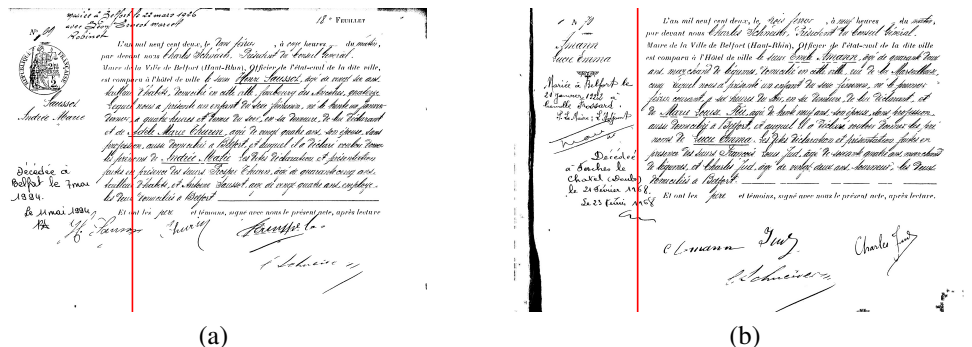


Figure 9: Bad side segmentation through histogram based paragraph detection that leads to data loss (Red: The segmentation position).

The same process can then be reused to find lines and words in sub-parts of the image. This method is capable of finding interesting parts with a small number of false positives, mainly on remaining noise and signatures as shown by figure 10. These artefacts are in various ways similar to text and could be avoided by tweaking preprocessing algorithm arguments. This strategy is, however, highly sensitive to changes in the structure and could not be used on various data. If the image is slightly rotated or present more or fewer features that were not planned, the segmentation will be completely deluded. This will therefore be only added to the dataset creation that allows cherry-picking by the user to train machine learning algorithms that are much more robust.

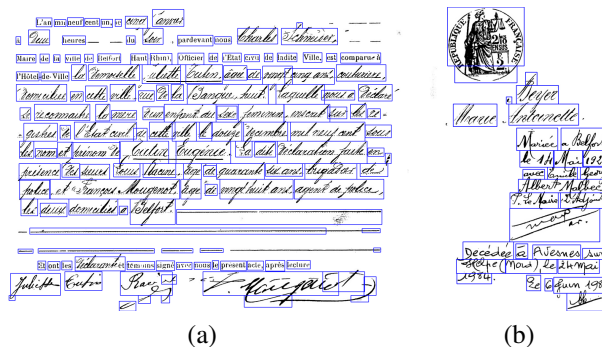
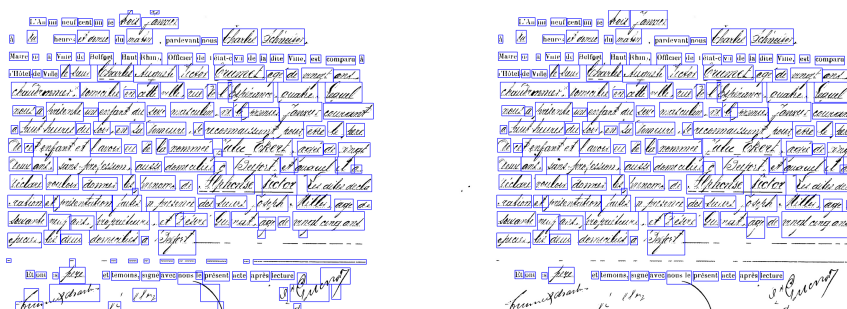


Figure 10: Word extraction results provided by the histogram based method

2.3.3 Bounding box edition tool

The accuracy provided by processes explained before is conclusive enough to use it as a way to speed up manual work. To provide these tools to contributors, a small and basic interface is used through which users can choose whether a bounding box should be kept, deleted or combined with others. This software allows the user to create the ground truth needed to train the neural network used for segmentation discussed in 2.3.5, without having to place all the bounding box manually but by correcting a segmentation. As highlighted in figure 11, the user has improved a lot the quality of the segmentation mainly on the signature part, false-positive lines and finally large handwritten words that were cut into parts.



(a) Histogram projection based segmentation output

(b) Bounding box edition tool output

Figure 11: Visualisation of bounding box changes after the use of the edition tool

2.3.4 Data augmentation

After correcting the bounding box detected using the tool presented in 2.3.3 and acquiring the ground truth needed to train the network, a data augmentation process has been set up. This method randomly selects some words that have already been segmented and randomly applies transformations, detailed in figure 12, to these images.

Augmentation type	Transformation parameters	Application probability
Resize	size $\in [0, 350]$	1.
Dilation	$I(x, y) \oplus \ker_s, s \in [3, 7]$	0.5
Salt and Pepper noise	$p_{\text{salt}}(x, y) = p_{\text{pepper}}(x, y) \in [0.005, 0.05]$	0.9

Figure 12: Data augmentation type and their probability

These words are then merged into a final image, an example of which is presented in figure 13, giving them random positions while making sure they do not overlap each other.

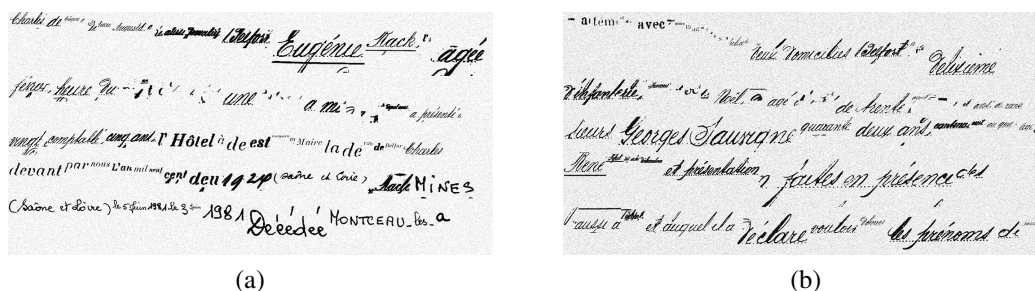


Figure 13: Examples of generated images

This automatic generation allows to have a wider range of data to use for network training and therefore have a better generalisation as shown by Wigington et al. [2017]. It presents, however, one major drawback which is including the errors of the first segmentation in the generated images.

2.3.5 EAST

As mentioned in the state of the art, the EAST network performs well at extracting text boxes from an image. In its original use, the network is trained to detect text in natural scenes, which means this dataset does not fit in our context. Nonetheless, the segmentation remains robust and accurate in diverse contexts. For this reason, adapting it to our data helps to generalize the results obtained by the correction performed by the user. In order to ease the task of the network, we chose to provide keep the histogram-based segmentation method to extract paragraph and then use the deep learning algorithm to detect word bounding box. This aims to attenuate the differences between ground truth data which can be significant due to the different writing.

Thus, to gather a sufficient amount of data to train the model for our context, we used the best results of the corrected histogram-based word level bounding-box. After training the model on this personal dataset, it can correctly detect the text inside the main features of the paragraphs given to it.

To finally provide extracted samples in a logically correct order, the detected bounding box set needs to be ordered based on the text layout. A naive sorting algorithm could be confused by uneven bounding boxes because of large capital letters or even badly cropped paragraphs. This problem is addressed by the algorithm 1. The latter's goal is to provide a top to bottom and left to right sort based on important changes within the bounding box height position distribution to

detect line breaks. Once this information has been acquired and a proper vertical sort performed, it only remains to sort horizontally the bounding box.

Algorithm 1: Bounding Box sorting algorithm

Input: EAST detected bounding boxes of size N

Output: Sorted bounding boxes

Sort Input content based on vertical coordinates

Distances: Array which contains the differences of the height between each successive bounding boxes

for $i = 1 \dots N$ **do**

$\alpha \leftarrow$ Mean height of input($i - 1$)

$\beta \leftarrow$ Mean height of input(i)

$\delta \leftarrow |\alpha - \beta|$

 Add δ to Distances

$\sigma \leftarrow \sqrt{\frac{1}{n} \sum_{k=0}^{n-1} (\text{Distances}_k - \overline{\text{Distances}})^2}$

$\sigma \leftarrow \max(0, \sigma)$

MaximasLocation \leftarrow Local maximas indices based on neighboring values and greater than σ

SortedBoxes \leftarrow First input's bounding box

CurrentRow: Array that will accumulate each bounding box of the same row

for $i = 1 \dots N$ **do**

if $i - 1 \in \text{MaximasLocation}$ **then**

 Sort CurrentRow content based on horizontal coordinates

 Add CurrentRow to SortedBoxes

 Clear CurrentRow

else

 Add i th input to CurrentRow

return SortedBoxes

By putting end to end the different tools described in this section, a process granting the ability to automate text-area detection is created in order to provide a text-focused image to OCR algorithms. We then computed the dice score of the histogram-based segmentation, EAST and the segmentation done by Tesseract for comparison purposes. It has been calculated over 29 different page fragments, using as ground truth the bounding boxes corrected by the user.

Method	Mean dice score for word segmentation \uparrow
Tesseract	54.36%
Histogram-based	62.17%
EAST	75.73%

Figure 14: Dice score for word segmentation, mean score on a test set of 29 fragments (\uparrow : Higher is better).

The results presented in figure 14 show that the segmentation performed by EAST is the closest to the corrected segmentation realised by the user. This means that, EAST-based segmentation is much more robust than the histogram-based method and then once integrated within the pipeline, will ease the HTR process.

2.4 Optical character recognition

2.4.1 Transcription dataset builder tool

Once the text segmentation is computed, it remains to acquire, in an accurate manner, the text data via an OCR process. This is done based on deep learning algorithms, using a gated convolutional neural network as described by de Sousa Neto et al. [2020]. Therefore having ground

truth data through which the network can be trained was necessary. Contributors were asked to manually annotate original segmented paper words. To reduce these interventions, the building of a tool that automatically presents all paper fragments to the user has helped saving time.

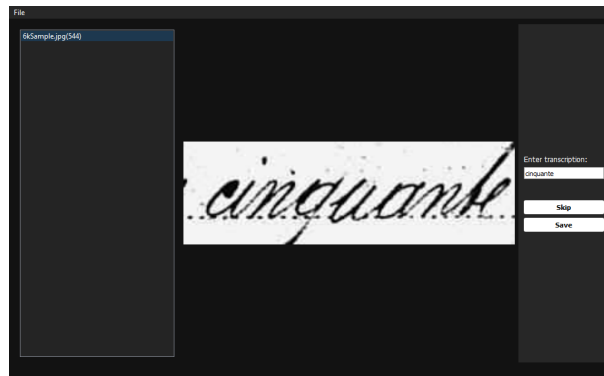


Figure 15: Transcription dataset builder tool screenshot.

The GUI, shown in figure 15, developed for this project aims to prevent time-consuming operation with the help of keyboard shortcuts that help the user to navigate in the segmented data and provide transcription. By avoiding the use of the mouse, confirmed users can benefit from a more natural experience and build a fairly substantial database that can directly be used to train HTR algorithms.

2.4.2 Handwritten text recognition

After transcribing 8,741 words, we created training, validation and test sets, which will then be used to train the network. The distribution between the different sets is the following : Train = 5,664 words, Validation = 1,535 words, Test = 1,542 words.

Once the model is trained on the data created, it can be used to extract the text from the images given in input. Then the whole process can be set up. It will take a full page as input and segment it into words on the basis of the same preprocess as the dataset creation part. By doing so, the images are then in the exact same format as the ones used for training. The words extracted are then concatenated following the text structure to have the different paragraphs well separated.

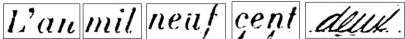

Original image	OCR results
	lan mil neuf cent deux
	Charles Schneider President du Conseil General

Figure 16: Examples of results obtained by the text recognition process

2.5 Experimental results

The number of detected words in every paragraph is then used to estimate the accuracy gain between both segmentation methods. The need for a comparison basis is, however, unavoidable concerning the metric meaning. The lack of perfectly accurate ground truth built through manual work in this project made us use the data partially corrected by contributors. This can be seen in the figure 17 that presents algorithms accuracy compared to the original one.

Method	Mean error rate in detected words count ↓
Histogram based	12.08
EAST	4.03

Figure 17: Text segmentation method metrics, mean error rate in words count on a test set of 144 paragraphs. (↓: Lower is better)

The error rate between ground truth and method results may be explained in different ways. First of all, the histogram-based method is the basis of the training data that contributors have sorted out to remove false positive from the dataset, which causes this method to have more detected words than the final dataset. The results provided by the EAST detector come from the signatures false positive detection as well as words bounding box inaccuracy. Indeed, the detector main issue is in handwritten text width inaccuracy that causes words to be mixed.

To have a global idea of how the whole process performs, the mean character error rate (CER) and the mean word error rate (WER) were used. To calculate these metrics the Levenshtein distance was evaluated between ground truth and the predictions on 11 pages that were not in the dataset. We also compared these results to the ones obtained using Tesseract.

Method	Punctuation & accentuation	Text part	Mean CER ↓	Mean WER ↓
Full text	Sensitive	Tesseract	69.68%	90.11%
		Ours	49%	80.31%
	Insensitive	Tesseract	68.2%	88.78%
		Ours	45.06%	70.81%
Main Paragraph	Sensitive	Tesseract	63.64%	84.28%
		Ours	30.93%	68.17%
	Insensitive	Tesseract	61.94%	82.1%
		Ours	25.62%	50.27%

Figure 18: CER and WER of the full process on 11 samples (↓: Lower is better).

The results shown in figure 18 reveal that the solution presented gives better results on the main paragraphs than on the full text. These paragraphs are easier to segment since no additional information has been added to them over time, therefore their shape is similar for all of them that make it easier to analyse. This points out that the layout analysis still needs to be improved so that all the additional data about one person remains together and is not mixed with the main paragraph. The word segmentation would also require improvement to avoid cut within words or skipped one in this process.

The resulted text of the presented pipeline is not fully accurate and yet insufficient to study the data extracted.

III CONCLUSION / FUTURE WORK

The proof of concept program developed for this project is at an early stage and should be upgraded to provide more features to final users.

First of all, the segmentation of paragraphs, based on histogram projection, is highly weak on layout changes and could be enhanced by the use of tools that are more flexible when dealing with paragraph segmentation as the one presented by Reul et al. [2017]. The use of deep learning solutions that allow such detection as shown by Chen and Seuret [2017] could also improve the performance of the process. This would ensure to reduce the segmentation error rate and make it more robust than the current segmentation strategy explained in 2.3.5. It may also help to segment paragraph more logically as it could provide less linear-based paragraph segmentation. Finally, this method might ease the adaptation of the current pipeline on other types of data and thus make it more flexible. The development of a fully modular software could also enhance the processing pipeline and make it more robust for new data shapes integration. This would ease the use by other teams and make it more accessible.

Then, as described in 2.1, the goal would be to get a fully-featured pipeline through which the user would almost only have to provide scanned images to acquire data in a common database format. These features might be developed based on NLP algorithms that could provide an accurate and deep analysis of text information that may help to create queryable data. This could then be used directly by final users through standard software.

The work presented in this paper aimed to propose a complete software pipeline through which users will be able to save time-based on state-of-the-art and production-ready algorithms. The choice of machine learning algorithms for both segmentation and recognition processes brings a general accuracy and robustness to the process which, however, requires ground truth datasets. Therefore, the creation of the latter, in an accurate and typical manner, needs manual intervention which has been strongly reduced by the help of temporary tools based on simple heuristics and traditional image processing strategy. The support of a basic user interface helped, moreover, the user throughout the validation of the pre-annotated data. These additional tools mainly saved time and allowed the creation of a sufficiently exhaustive dataset by small teams in a limited period. This makes it more accessible and less expensive to set up separately from the project size and its number of contributors.

The use of state-of-the-art algorithms makes the full process more sustainable and robust. This pipeline can then be made suitable for new data within a small amount of time and without the need for a large contributors number. This work is a basis that could be enhanced across the addition of natural language processing algorithms that will, afterwards, be useful to analyse the recognised text and create queryable data that might directly be employed at the end of the treatment chain.

References

- Alkalai and Sorge. A Histogram-Based Approach to Mathematical Line Segmentation. In José Ruiz-Shulcloper and Gabriella Sanniti di Baja, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, pages 447–455, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-41822-8. doi: 10.1007/978-3-642-41822-8_56.
- Bataineh, Abdullah, and Omar. An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recognition Letters*, 32:1805–1813, October 2011. doi: 10.1016/j.patrec.2011.08.001.
- Thomas M. Breuel, Adnan UI-Hasan, Mayce Ali Al-Azawi, and Faisal Shafait. High-Performance OCR for Printed English and Fraktur Using LSTM Networks. In *2013 12th International Conference on Document Analysis and Recognition*, pages 683–687, August 2013. doi: 10.1109/ICDAR.2013.140. ISSN: 2379-2140.
- Chen and Seuret. Convolutional Neural Networks for Page Segmentation of Historical Document Images. *arXiv e-print*, April 2017.
- de Sousa Neto, Bezerra, Toselli, and Lima. Htr-flor: A deep learning system for offline handwritten text recognition. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 54–61, 2020. doi: 10.1109/SIBGRAPI51738.2020.00016.
- Dudhabaware and Madankar. Review on natural language processing tasks for text documents. In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–5, 2014. doi: 10.1109/ICCIC.2014.7238427.
- Fischer, Indermühle, Bunke, Viehhauser, and Stolz. Ground truth creation for handwriting recognition in historical documents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS '10*, page 3–10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587738. doi: 10.1145/1815330.1815331.
- Gatos, Pratikakis, and Perantonis. *An Adaptive Binarization Technique for Low Quality Historical Documents*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28640-0.
- Granet, Morin, Mouchère, Quiniou, and Viard-Gaudin. Transfer Learning for Handwriting Recognition on Historical Documents. In *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Madeira, Portugal, January 2018.
- Haton. La longévité des Belfortain.e.s né.e.s en 1905 et 1906: une analyse multivariée/ The longevity of the inhabitants of Belfort born in 1905-1906: a multivariate analysis. UTBM unpublished working paper, 2020.
- Heyberger. L’industrialisation de Belfort : une conséquence positive du siège de 1870-1871 ? Approche par l’histoire anthropométrique. In *Robert Belot (ed.) 1870. De la guerre à la paix. Strasbourg-Belfort.*, pages 207–17. Hermann, 2013.
- Huang and Srihari. Word segmentation of off-line handwritten documents. In *Electronic Imaging*, 2008.
- R. Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C. Popat. A Scalable Handwritten Text Recognition System. *arXiv e-print*, June 2019.
- Massot, Sforzini, and Ventresque. Transcribing Foucault’s handwriting with Transkribus. *Journal of Data Mining and Digital Humanities*, Atelier Digit_Hum, March 2019. Publisher: Episciences.org.
- Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. ISSN 2168-2909. doi: 10.1109/TSMC.1979.4310076. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics.
- Rakshit and Basu. Development of a multi-user handwriting recognition system using tesseract open source OCR engine. *CoRR*, abs/1003.5886, 2010.
- Rakshit, Basu, and Ikeda. Recognition of handwritten textual annotations using tesseract open source ocr engine for information just in time (ijit), 2010.
- Renton, Chatelain, Adam, Kermorvant, and Paquet. Handwritten text line segmentation using fully convolutional network. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 05, pages 5–9, 2017. doi: 10.1109/ICDAR.2017.321.
- Christian Reul, Uwe Springmann, and Frank Puppe. LAREX: A Semi-automatic Open-source Tool for Layout Analysis and Region Extraction on Early Printed Books. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage, DATECH2017*, pages 137–142, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5265-9. doi: 10.1145/3078081.3078097.
- Verónica Romero, Nicolás Serrano, Alejandro H. Toselli, Joan Andreu Sánchez, and Enrique Vidal. Handwritten Text Recognition for Historical Documents. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 90–96, Hissar, Bulgaria, September 2011. Association for Computational Linguistics.
- Schlagdenhauffen. Optical Recognition Assisted Transcription with Transkribus: The Experiment concerning Eugène Wilhelm’s Personal Diary (1885-1951). *Journal of Data Mining and Digital Humanities*, Atelier

Digit_Hum, August 2020.

- R. Smith. An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, September 2007. doi: 10.1109/ICDAR.2007.4376991. ISSN: 2379-2140.
- Curtis Wigington, Seth Stewart, Brian Davis, Bill Barrett, Brian Price, and Scott Cohen. Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 639–645, November 2017. doi: 10.1109/ICDAR.2017.110. ISSN: 2379-2140.
- C. Wolf, J.-M. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Object recognition supported by user interaction for service robots*, volume 2, pages 1037–1040, Quebec City, Que., Canada, 2002. IEEE Comput. Soc. doi: 10.1109/ICPR.2002.1048482.
- Mohamed Yousef and Tom E. Bishop. Origaminet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- Zhou, Yao, Wen, Wang, Zhou, He, and Liang. East: An efficient and accurate scene text detector, 2017.
- Zohrevand, Sadri, Imani, and Yeganezad. Line segmentation in persian handwritten documents based on a novel projection histogram method. In *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 70–74, 2019. doi: 10.1109/PRIA.2019.8786006.