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

Cyprien Plateau-Holleville, Enzo Bonnot, Franck Gechter, Laurent Heyberger

Journal of Data Mining and Digital Humanities

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]

[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 of contributors transcription teams of sufficient size can thus significantly slow down linked research work and make its cost dramatically rise. Developing automation through OCR (Optical Character Recognition), HTR (Handwritten Text Recognition), and NLP (Natural Language Processing) 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 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 in order 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 Some fields such as historical document analysis are, however, still challenging even with the growth of these technologies.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 engineering sciences and human and social sciences entitled "Techn'Hom Time Machine" (THTM). Led by researchers from the University of Technology of Belfort-Montbeliard, 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]. The use of the vital records, 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.

Data provided in this project is composed of unlabelled-scanned images of French vital records as shown in figure 1. Each sample is divided into four parts itself giving information about one person. These are built on the basis of 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.

This type of problem is mainly solved by time-consuming manual work possibly 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 closed-source HTR engine that can make it much less accessible. Other well-known software such as Tesseract offers well-performing 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 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 aims to limit at most manual interventions.

The rest of the paper is structured as follows. The first section is focused on historical data analysis state of the art on the basis of image processing and deep learning algorithms. The proposition details are then explained in the second section while in the third we discuss and show our experimental results. Finally, the fourth section concludes the paper in providing some potential future works.

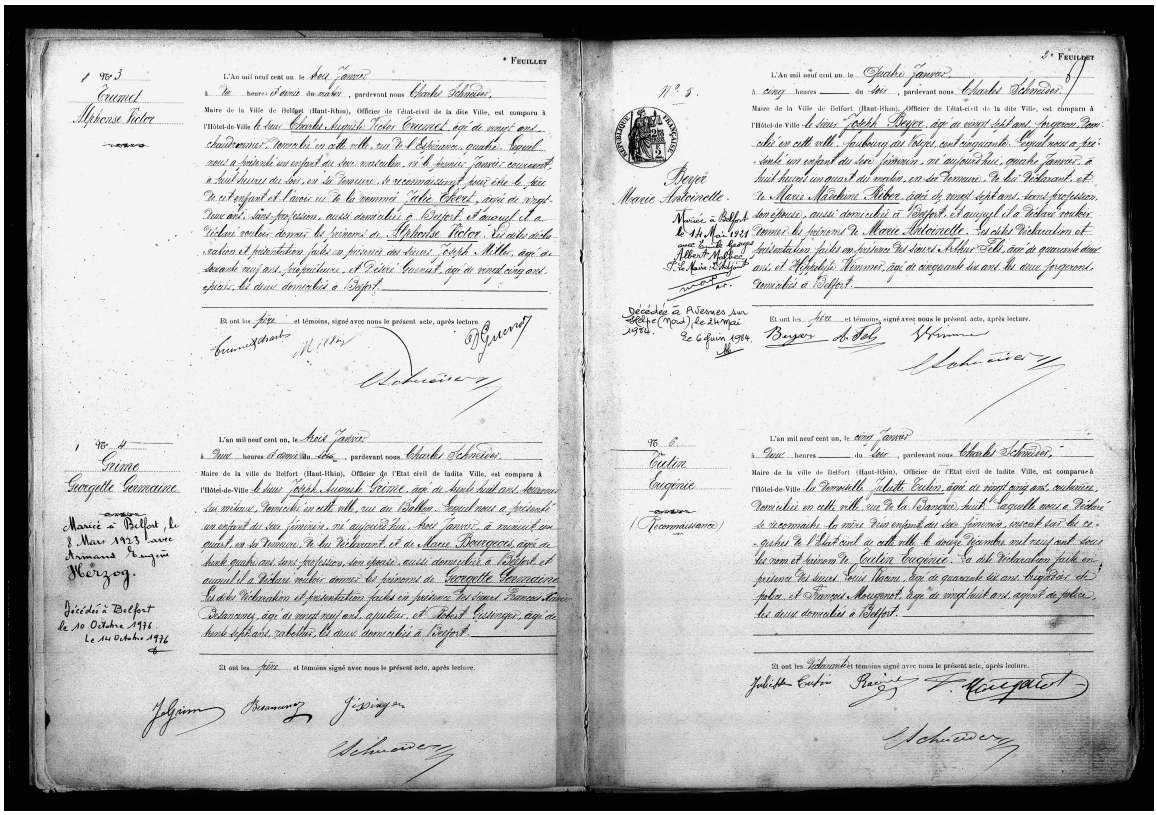


Figure 1: Scanned vital record sample (registres d'état-civil des naissances, 1901, archives of the "Territoire de Belfort" department)

I STATE OF THE ART

1.1 Automatic Text Processing

1.1.1 Segmentation

Handwritten text segmentation is the first step of every handwritten text recognition process. 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 machine learning based approach and especially convolutional networks allowing to get better result with a much more stable process. Renton et al. [2017] presented their solution using these technologies which gives 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 inaccuracies in the first step will be amplified in the second one. 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 natural scene or complex background, hence adapting it to the context of these vital records would be a suitable option.

1.1.2 Recognition

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. One of the main subfield of OCR is HTR which is very complex due to the fact that 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 establish manually 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. 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. 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.

1.1.3 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 an 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-data-based datasets in order 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 2. This article will only present our work on the text segmentation and the text recognition parts of the pipeline.

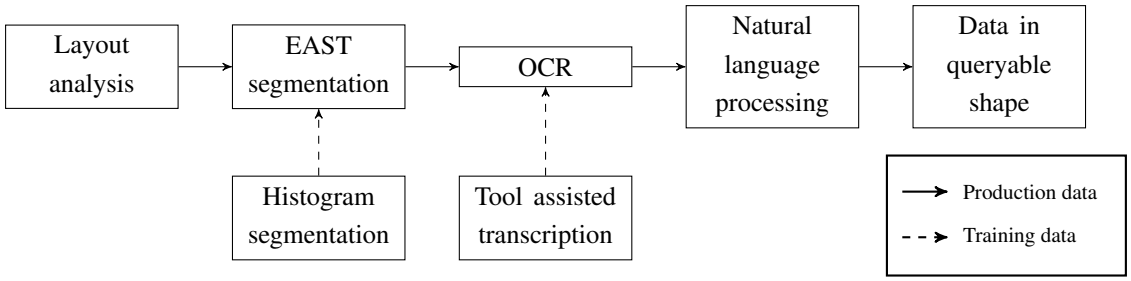


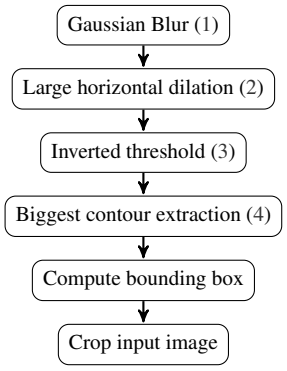
Figure 2: Data acquisition pipeline

2.2 Text Segmentation

As previously explained, the text segmentation is divided into two parts, one for the training data creation process, the second for the production pipeline process. The need of 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, with a small workforce this task can be highly time consuming. The creation of dedicated tools aiming to help this task is a good trade-off that enables strong time savings and requires a lot less contributors in order to establish sets that will, then, enable accurate problem solving. This is also a valid solution for the segmentation since text words bounding box annotation can be redundant and error-prone for the contributor. This part will present the created tools that aims to make the annotation steps easier and the way in which it interacts with the learning phase.

2.2.1 Preprocessing

The input data of the process is raw scanned grayscale images containing scanner artifacts and frames. This initial condition makes segmentation process harder due to background noises. Therefore, several preprocessing steps are applied whose goal is to enhance text while reducing at most non-text parts. In order to achieve this, the use of the background detection based on the process described in figure 3 allows to find the document outline in a robust way on our dataset. Then, a gamma correction is performed to improve the general luminosity of the image.



$$G(x,y)=\frac{1}{2\pi\sigma^2}.e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$I(x,y)=I(x,y)*G(x,y) \quad (1)$$

$$B(x,y)=(u_1 \quad u_2 \quad \dots \quad u_N), \forall i \in [1,N], u_i=1$$

$$I(x,y)=I(x,y)\oplus B(x,y) \quad (2)$$

$$T(I(x,y),t)=\begin{cases}1, if 1-I(x,y)>t\\0, otherwise\end{cases} \quad (3)$$

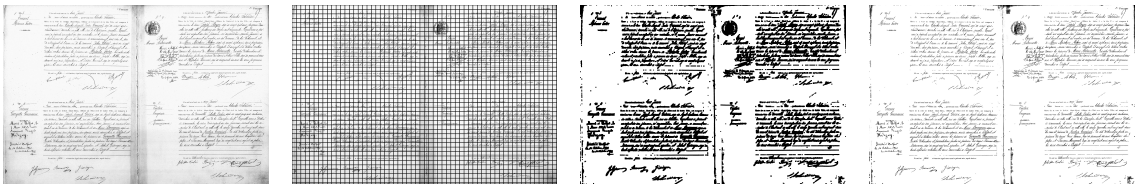
$$contour=max(area(contours)) \quad (4)$$

Figure 3: Scanner background extraction process

Finally, the classification of both text and background is made through a segmentation 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 case[1] in a similar way to Bataineh et al. [2011] and Gatos et al. [2004]. As can be seen in figure 4, the main property of this method is to apply threshold on image parts 4b, that are then used to build a mask 4c, which finally allows to segment interesting information of the image 4d. Its properties are particularly well suited to text segmentation since the contrast of the image can be variable, which would cause parts of the text to be wiped out by the classical threshold, whereas this method mainly allows the erasure of paper-related data and noise, while retaining mainly textual information as can be seen in the binarisation process in figure 4. The quality of the segmentation is therefore

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

less linked to the complete image characteristic but rather to local information which provide accurate and robust results throughout its use on our dataset.



(a) Input image without background

(b) Median threshold blocks

(c) Dilated mask

(d) Output image

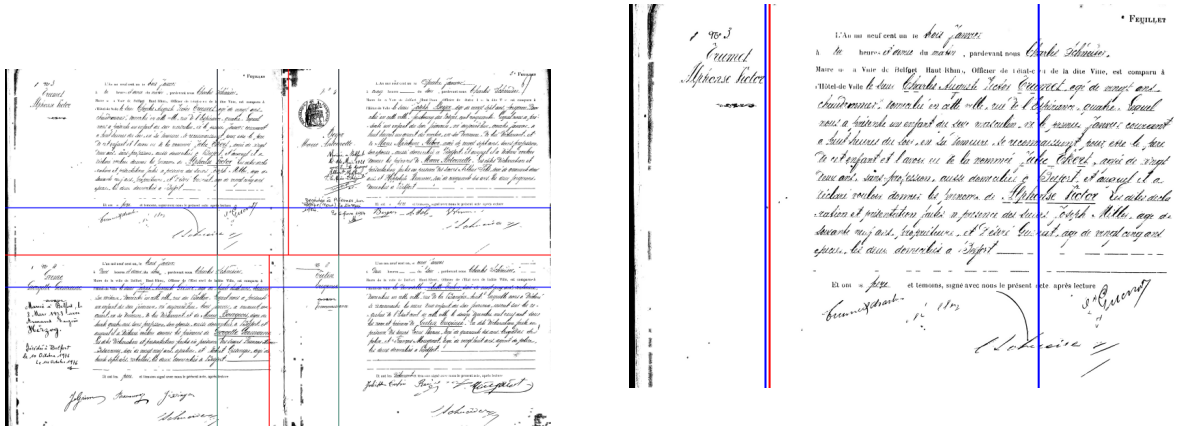
Figure 4: Adaptive local binarization process

This method requires, however, to parameterise the size of the blocks that leads it to be less robust to strong image changes. The latter would require specific tests in order to properly set the values.

2.2.2 Histogram projection based segmentation

Once the preprocessing steps are applied to the image, the enhancement provided allows to decrease a lot the analysis complexity. 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 might only require small tweaks during manual inspections. However, heuristics developed for this part can be 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. Moreover, this can be changed for other use cases without major pipeline transformation.

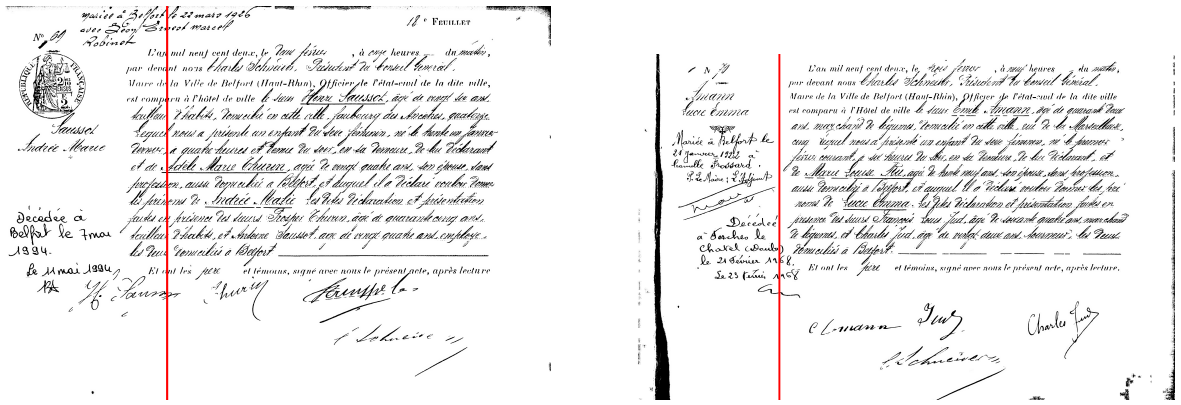
Thanks to preprocessing quality and noise removal, histogram projections can be used for the purpose of text segmentation as shown by Alkalai and Sorge [2013]. The latter demonstrate how, in the case of backgroundless images, text lines can be analysed in a really accurate way through its vertical and horizontal histogram properties. Zohrevand et al. [2019] validated this method as well with the support of deskew algorithms which is not needed in the current dataset since lines generally have horizontal levels. Text area's main shapes are then found based on white and black pixel distribution. This method is, 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 area derived from the most common layout as we can see in figure 5. The selection of the less significant element provide finally a way to find paragraph borders.



(a) Layout segmentation (b) Side segmentation

Figure 5: Visualisation of the histogram projection based segmentation (Red : Selected local minimas. Blue, green and black : Histogram projection bounds)

As shown in figure 5b, 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 6.



(a) (b)

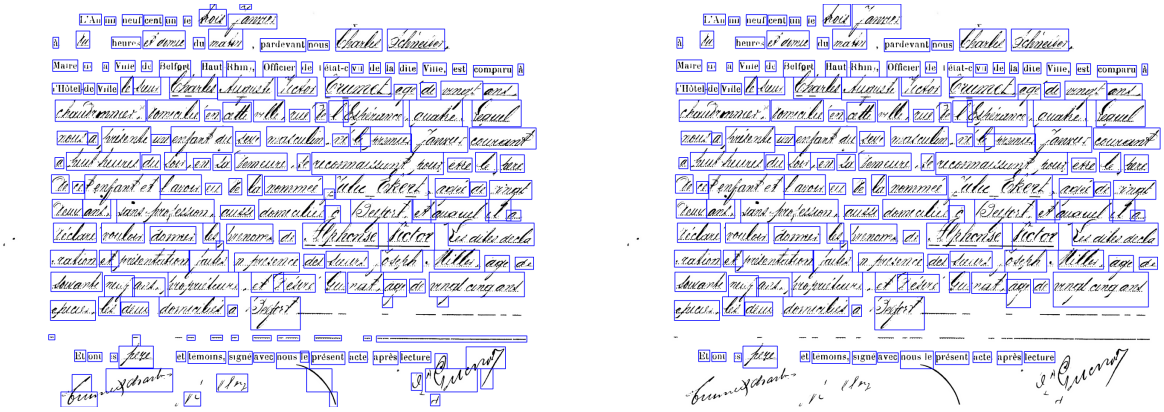
Figure 6: Bad side segmentation through histogram based paragraph detection that leads to data loss (Red: The segmentation position).

The same process can then be reused in order to find lines and words in sub parts of the image. This method is especially capable of finding interesting parts with a small number of false positives, mainly on remaining noise and signatures. The latter two 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. If the image is slightly rotated or present more or less features that was not planned, the segmentation will be completely deluded. This will therefore be only added to the dataset creation tool that allows cherry picking by the user in order to train machine learning algorithms that are much more robust.

2.2.3 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. In order 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 enables to improve a lot the quality of the dataset across a really small amount of time without a large number of contributors. As highlighted in figure 7, 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

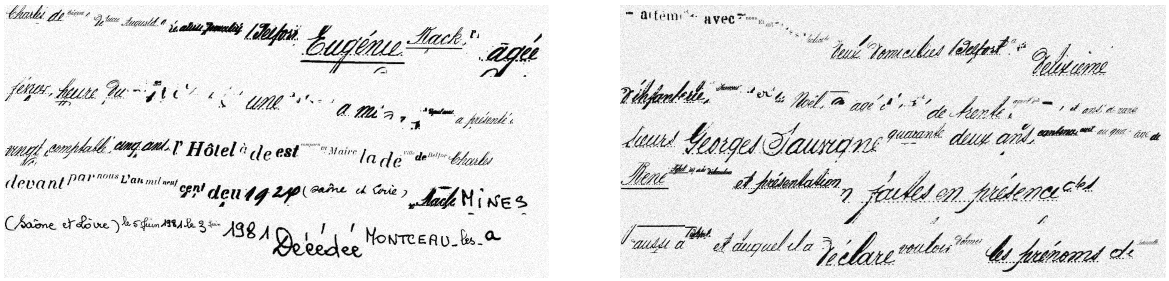
(b) Bounding box edition tool output

Figure 7: Visualisation of bounding box changes after the use of the edition tool

Within a few hours of use, a team of two contributors fixed more than a hundred of full samples which could then be directly used to train the machine learning algorithms, that would have taken several days of work without the support of this software.

2.2.4 Data augmentation

After correcting the bounding box detected using the tool presented in 2.2.3 and in order to acquire more data to train the network without spending more time, a data augmentation process has been set up. This method picks randomly some words that have already been segmented and applies transformations (rotation, rescale...) to these images. Then these words are assembled into one final image, giving them random positions while making sure they do not overlap each other. Finally, some noise is added to the background so that the data is varied. The results can be seen in figure 8.



(a) (b)

Figure 8: Examples of generated images

This automatic generation allows to have a much wider range of data to use for network training and therefore have a better generalisation. However, this method presents one major drawback which is including the errors of the first segmentation in the generated images.

2.2.5 *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 obtain good results for the majority of the pages where the histogram-based method is more unstable. However, changes throughout ground truth data can be significant due to the different writing. This is attenuated with histogram-based segmentation for paragraphs only. The latter also helps to preserve the semantics of the text between paragraphs before OCR.

Thus, in order to gather a sufficient amount of data to train the model for our context, we used the best results of the histogram-based segmentation, corrected by a human and formatted to something understandable by the network. After training the model on this personal dataset, it is able to correctly detect the text inside the main features of the paragraphs given to it

To finally provide the text in a semantically correct order, the detected bounding box set needs to be ordered on the basis of the text layout as presented in the algorithm 1. The latter sort bounding boxes from top to bottom and from left to right since it is the writing style used in the data of this project.

Algorithm 1: Bounding Box sorting algorithm

Input: EAST detected bounding boxes
Output: Sorted bounding boxes
input ← sort(input, axis=Vertical)
d ← []
for $i \leftarrow 1 \dots$ length(input) **do**
    fHeightCenter ← $\frac{\text{input[i - 1].y1+input[i - 1].y2}}{2}$
    sHeightCenter ← $\frac{\text{input[i].y1+input[i].y2}}{2}$
    cDiff ← |fHeightCenter − sHeightCenter|
    d ← [d; cDiff]
$\sigma \leftarrow \sqrt{\frac{1}{n}\sum_{k=0}^{n-1}\left(d_k - \bar{d}\right)}$
$\sigma \leftarrow \max(0, \sigma)$
maximasIndices ← findPeaks(d, height=σ)

sortedBoxes ← [input[0]]
currentRow ← []
for $i \leftarrow 0 \dots$ length(input) **do**
    `// On new text line`
    if $i - 1 \in$ maximasIndices **then**
        currentRow ← sort(currentRow, axis=Horizontal)
        sortedBoxes ← [sortedBoxes; currentRow$_0$; . . . ; currentRow$_n$]
        currentRow ← [input[i]]
    else
        currentRow ← [currentRow; input[i]]
return [sortedBoxes; currentRow$_0$; . . . ; currentRow$_n$]

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 text-focused image to OCR algorithms.

2.3 Optical character recognition

2.3.1 Transcription dataset builder tool

Once the text segmentation computed, it remains to acquire, in an accurate manner, the text data via an OCR process. This is done on the basis of deep learning algorithms that needs ground truth data through which it can be trained. Contributors were asked to annotate, manually, original segmented paper words. In order to reduce these interventions, the building of a tool that automatically presents all paper fragments to the user has helped saving time.

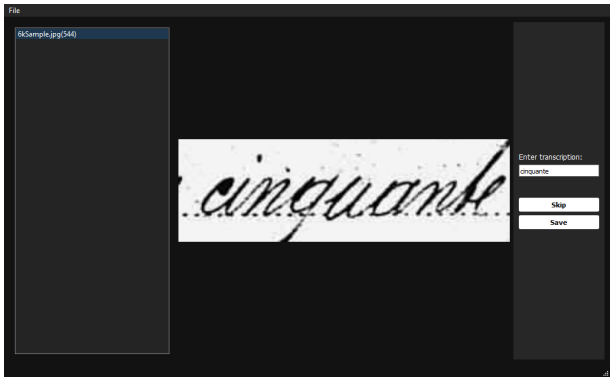


Figure 9: Transcription dataset builder tool screenshot.

The GUI, shown in figure 9, 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 are then able to save time and build a fairly substantial database than can directly be used to train HTR algorithms.

2.3.2 Handwritten text recognition

After transcribing a sufficient amount of data, it is then transformed into training, validation and test sets resulting in the distribution shown figure 10, which will then be used to train the network.

Dataset partition	**Amount**
Train	5,664
Validation	1,535
Test	1,542

Figure 10: Data distribution between the different batches

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. By doing so, the images are formatted like the ones used for training. The words extracted are then concatenated following the text structure in order to have the different paragraphs well separated.

III EXPERIMENTAL RESULTS

The number of detected words in every paragraph is then used to estimate the accuracy gain between both segmentation methods. The need of 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 11 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 11: Text segmentation method metrics (Mean error rate in words count on a test set of 144 paragraphs)

The error rate between ground truth and method results may be explained through different ways. First of all, the histogram-based method is the basis of the training data that contributors have sorted out in order 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 together.

In order to have a global idea on 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.

Punctuation & accentuation	**Text part**	**Mean CER**	**Mean WER**
Sensitive	Full text	49%	80.31%
	Main paragraph	30.93%	68.17%
Insensitive	Full text	45.06%	70.81%
	Main paragraph	25.62%	50.27%

Figure 12: CER and WER of the full process on 11 samples

The results shown in figure 12 reveal that the solution presented gives better results on the main paragraphs than on the full text. These paragraphs are easier to segment due to the fact that no additional information has been added to them over time, therefore their shape is really 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 in order 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.

3.1 Discussion

The proof of concept program developed for this project is at an early stage and should be upgraded in order 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 should be enhanced by the use of deep learning solutions that allow such detection as shown by Chen and Seuret [2017]. This would ensure to reduce the segmentation error rate and make it more robust than the current segmentation strategy explained in 2.2.5. It could also help to segment paragraph in a semantically more correct manner as it could provide sensible paragraph segmentation less linear-based. 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 shown on the full pipeline, the goal would be to get a fully featured pipeline through which the user would almost only have to provide scanned images in order to acquire data in a common database format. These features might be developed on the basis of NLP algorithms that could provide an accurate and deep analysis of text information that might help to create queryable data, which would be used directly by final users through standard software.

IV CONCLUSION

The work presented in this paper aimed to propose a fully featured software pipeline through which users will be able to save time on the basis of 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 is strongly reduced by the help of temporary tools based on simple heuristics and traditional image processing strategy. The support of a basic user interface helps, moreover, the user throughout the validation of the pre-annotated data. These additional tools mainly save time and allow the creation of a sufficiently exhaustive dataset by small teams in a limited period of time. 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 to new data type. This pipeline can then be made suitable for the latter within a small amount of time and without the need of large contributors size.

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.

Chen and Seuret. Convolutional Neural Networks for Page Segmentation of Historical Document Images. *arXiv:1704.01474 [cs, stat]*, April 2017. arXiv: 1704.01474.

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. pages 207–17. Hermann, robert belot edition, 2013.

Huang and Srihari. Word segmentation of off-line handwritten documents. In *Electronic Imaging*, 2008.

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.

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.

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.

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.